

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-110844

(43)Date of publication of application : 22.04.1994

(51)Int.Cl.

G06F 15/16

G06F 12/08

(21)Application number : 05-199772

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 11.08.1993

(72)Inventor : IWASA SHIGEAKI
OMIZO TAKASHI

(30)Priority

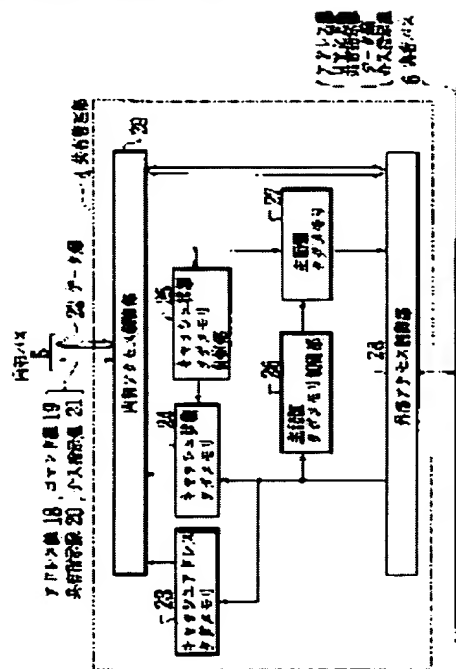
Priority number : 04213133 Priority date : 11.08.1992 Priority country : JP

(54) DECENTRALIZED SHARED MEMORY TYPE MULTIPROCESSOR SYSTEM

(57)Abstract:

PURPOSE: To reduce the traffic of a shared bus and to hold the consistency of a cache in a decentralized shared memory type multiprocessor system consisting of plural processors containing the CPUs, the cache memories, and the partial and shared control parts of a main storage which are connected together via the shared bus.

CONSTITUTION: A shared control part 4 includes a main storage tag memory 27 which stores the information on a fact whether the address of its own main memory is stored (shared) in the cache memory of another processor or not (non-shared) and a cache state tag memory 24 which stores the information on a fact whether the data entry of the cache memory is updated and rewritten (dirty) in a shared memory or not (clean). Then a CPU gives an invalidating instruction onto a shared bus to assure the cache consistency as long as a main storage tag is shared when the CPU writes the information in its own main memory. Meanwhile the CPU gives nothing to the shared bus if the main storage tag is not shared and performs the processing in its own processor only. If the cache state tag is dirty to the access request given from another processor unit, the cache consistency is held via the access request.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than
the examiner's decision of rejection or
application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of
rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12)公開特許公報 (A)

(11)特許出願公開番号

特開平6-110844

(43)公開日 平成6年 (1994) 4月22日

(51)Int. Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 2 0 K	8840-5L		
12/08	3 1 0 B	7608-5B		

審査請求 未請求 請求項の数5 (全 64 頁)

(21)出願番号 特願平5-199772

(22)出願日 平成5年 (1993) 8月11日

(31)優先権主張番号 特願平4-213133

(32)優先日 平4 (1992) 8月11日

(33)優先権主張国 日本 (J P)

(71)出願人 000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地

(72)発明者 岩佐 繁明
神奈川県川崎市幸区小向東芝町1 株式会
社東芝研究開発センター内

(72)発明者 大溝 孝
神奈川県川崎市幸区小向東芝町1 株式会
社東芝研究開発センター内

(74)代理人 弁理士 三好 秀和 (外3名)

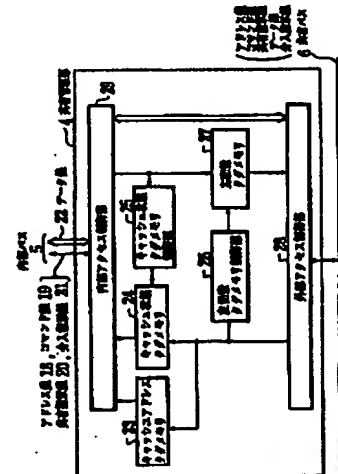
(54)【発明の名称】 分散共有メモリ型マルチプロセッサシステム

(57)【要約】

【目的】 CPUとキャッシュメモリと主記憶の一部と共有管理部とを備えた複数のプロセッサを共有バスで接続して成る分散共有メモリ型マルチプロセッサシステムにおいて、共有バス上のトラフィックを少なくして、キャッシュの一貫性保持を実現する。

【構成】 共有管理部4が、自己の主メモリのアドレスが他のプロセッサのキャッシュメモリ上に存在するか

(共有か) 否か (非共有か) の情報を記憶する主記憶タグメモリ27やキャッシュメモリのデータ・エントリが更新され且つ共有メモリに書き戻されていないか (ダーティ) 否か (クリーン) の情報を記憶するキャッシュ状態タグメモリ24備え、CPUが自己の主メモリへの書き込みを行う際に、主記憶タグが共有であれば共有バス上に無効化命令を出してキャッシュの一貫性を保証し、非共有であれば共有バス上には何も出さずに自己のプロセッサ内部だけで処理を行う。又、別のプロセッサ・ユニットからのアクセス要求に対し、キャッシュ状態タグがダーティであればこのアクセス要求に介入して一貫性保持を行う。



1

【特許請求の範囲】

【請求項1】 共有バスを介して結合された複数のプロセッサ・ユニットから成る分散共有メモリ型マルチプロセッサ・システムであって、各プロセッサ・ユニットが、

CPUと、

内部バスを介してCPUと接続され、システムの共有メモリの分散された一部を記憶する主メモリと、

内部バスを介して主メモリと接続され、共有メモリの選択されたデータ・エントリをキャッシュするCPUキャッシュメモリと、

内部バスを介して主メモリとCPUキャッシュメモリと接続され、内部バスと共有バスをインタフェースする共有管理部で、

該各プロセッサ・ユニット上の主メモリの各データ・エントリが他のプロセッサ・ユニット上のCPUキャッシュメモリに共有されているか否かを示す主メモリの各データ・エントリの共有状態を記憶する主記憶タグ手段と、

CPUによる主メモリの或るデータ・エントリに対する書き込み命令の実行に対して、主記憶タグ手段の該或るデータ・エントリの共有状態が該或るデータ・エントリは他のプロセッサ・ユニット上のCPUキャッシュメモリに共有されていることを示している時に、該他のプロセッサ・ユニット上のCPUキャッシュメモリの該或るデータ・エントリを無効化するように該他のプロセッサ・ユニットに命令する無効化命令を共有バス上に発行する命令手段と、

CPUキャッシュメモリに記憶された選択されたデータ・エントリのアドレスを記憶するキャッシュ・タグ手段と、

某データ・エントリに対する無効化命令を共有バスから受け取り該某データ・エントリのアドレスがキャッシュ・タグ手段に記憶されている時に、CPUキャッシュメモリの該某データ・エントリを無効化する制御手段と、を含むものと、から成るシステム。

【請求項2】 共有バスを介して結合された複数のプロセッサ・ユニットから成る分散共有メモリ型マルチプロセッサ・システムであって、各プロセッサ・ユニットが、

CPUと、

内部バスを介してCPUと接続され、システムの共有メモリの分散された一部を記憶する主メモリと、

内部バスを介して主メモリと接続され、共有メモリの選択されたデータ・エントリをキャッシュするCPUキャッシュメモリと、

内部バスを介して主メモリとCPUキャッシュメモリと接続され、内部バスと共有バスをインタフェースする共有管理部で、CPUキャッシュメモリの各データ・エントリが該各プ

2

ロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないか否かを示すCPUキャッシュメモリの各データ・エントリのキャッシュ状態を記憶するキャッシュ状態タグ手段と、共有バスを介して受け取った別のプロセッサ・ユニットからの或るデータ・エントリに対するアクセス要求に対して、キャッシュ状態タグ手段の該或るデータ・エントリのキャッシュ状態が該或るデータ・エントリは該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないことを示している時に、該別のプロセッサ・ユニット上の主メモリの該或るデータ・エントリへのアクセスに介入する介入手段と、

介入手段が該別のプロセッサ上の主メモリの該或るデータ・エントリへのアクセスに介入した時に、キャッシュ状態タグ手段の該或るデータ・エントリのキャッシュ状態を、該或るデータ・エントリの所有権を持った一キャッシュ状態から該或るデータ・エントリの所有権を持たない他のキャッシュ状態に状態遷移させ、該各プロセッサ・ユニットからの某データ・エントリへのアクセスが他のプロセッサ・ユニットの介入手段により介入された時に、キャッシュ状態タグ手段の該某データ・エントリのキャッシュ状態を、該某データ・エントリの所有権を持たない一キャッシュ状態から該某データ・エントリの所有権を持った他のキャッシュ状態に状態遷移させるように制御する制御手段と、を含むものと、から成るシステム。

【請求項3】 共有バスを介して結合された複数のプロセッサ・ユニットからなる分散共有メモリ型マルチプロセッサ・システムであって、各プロセッサ・ユニットが、

CPUと、

内部バスを介してCPUと接続され、システムの共有メモリの分散された一部を記憶する主メモリと、

内部バスを介して主メモリと接続され、共有メモリの選択されたデータ・エントリをキャッシュするCPUキャッシュメモリと、

内部バスを介して主メモリとCPUキャッシュメモリと接続され、内部バスと共有バスをインタフェースする共有管理部で、

該各プロセッサ・ユニット上の主メモリの各データ・エントリが他のプロセッサ・ユニット上のCPUキャッシュメモリに共有されているか否かを示す主メモリの各データ・エントリの共有状態を記憶する主記憶タグ手段と、

CPUキャッシュメモリの各データ・エントリが該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないか否かを示すCPUキャッシュメモリの各データ・エントリのキャッシュ状態を記憶するキャッシュ状態タグ手段と、

CPUによる主メモリの或るデータ・エントリに対する読み出し／書き込み命令の実行に対して、主記憶タグ手段の該或るデータ・エントリの共有状態が該或るデータ・エントリは他のプロセッサ・ユニット上のCPUキャッシュメモリに共有されていることを示している時に、該他のプロセッサ・ユニット上のCPUキャッシュメモリの該或るデータ・エントリを共有バス上に読み出すように該他のプロセッサ・ユニットに命令する読み出し命令を共有バス上に発行する命令手段と、共有バスを介して受け取った別のプロセッサ・ユニットからの読み出し命令に対して、キャッシュ状態タグ手段の某データ・エントリのキャッシュ状態が該某データ・エントリは該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないことを示している時に、該別のプロセッサ・ユニット上の主メモリの該某データ・エントリへのアクセスに介入する介入手段と、を含むものと、から成るシステム。

【請求項4】 共有バスを介して結合された複数のプロセッサ・ユニットから成る分散共有メモリ型マルチプロセッサ・システムであって、各プロセッサ・ユニットが、

CPUと、

内部バスを介してCPUと接続され、システムの共有メモリの分散された一部を記憶する主メモリと、

内部バスを介して主メモリと接続され、共有メモリの選択されたデータ・エントリをキャッシュするCPUキャッシュメモリと、

内部バスを介して主メモリとCPUキャッシュメモリと接続され、内部バスと共有バスをインタフェースする共有管理部で、

CPUキャッシュメモリの各データ・エントリが該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないか否かを示すCPUキャッシュメモリの各データ・エントリのキャッシュ状態を記憶するキャッシュ状態タグ手段と、共有バスを介して受け取った別のプロセッサ・ユニットからの或るデータ・エントリに対するアクセス要求に対して、キャッシュ状態タグ手段の該或るデータ・エントリのキャッシュ状態が該或るデータ・エントリは該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないことを示している時に、該別のプロセッサ・ユニット上の主メモリの該或るデータ・エントリへのアクセスに介入する介入手段と、を含むものと、から成るシステム。

【請求項5】 請求項4記載のシステムにおいて、共有管理部が更に、

該各プロセッサ・ユニット上の主メモリの各データ・エントリの共有状態を記憶する主記憶タグ手段で、共有状態は該各データ・エントリの所有権が該各プロセッサ・

ユニット上に存在することを示す[H]状態と、該各データ・エントリの所有権が他のプロセッサ・ユニット上に存在することを示す[A]状態とを含むものと、CPUによる主メモリの某データ・エントリに対するアクセス要求に対して、主記憶タグ手段の該某データ・エントリの共有状態が[A]状態である時に、該他のプロセッサ・ユニット上のCPUキャッシュメモリの該某データ・エントリを共有バス上に読み出すように該他のプロセッサ・ユニットに命令する読み出し命令を共有バス上に発行する命令手段と、を含むシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は電子計算機、特に、複数のCPUをバスで結合する密結合マルチプロセッサシステムに関する。

【0002】

【従来の技術】 共有メモリ型マルチプロセッサは、図1のように複数のプロセッサ（プロセッサ1～プロセッサn）1000-1～1000-nと共有メモリ1000-1～1000-mをバスまたはスイッチを用いた結合路1002で結合した構成を持ち、メモリは全てのプロセッサから平等にアクセスすることができる。よって、並列プログラムは主メモリ上のメモリを区別することなくアクセスを行なうことができ、他のプロセスと共有している変数をアクセスする際に、そのプロセスがプロセッサ間に跨っているか否かを意識せずにアクセスを行なうことができる。

【0003】 ところで、並列プログラムといえども、共有変数へのアクセスばかりでプログラムが実行されるわけではない。むしろ、メモリアクセスの大部分はそのプログラムの局所変数へのアクセスであり、マルチプロセッサの性能を高める上で、局所変数のアクセス性能が大きな影響を及ぼす。

【0004】 図1の共有メモリ型マルチプロセッサではプロセッサで共有していない局所変数へのアクセスであっても、キャッシュミスを起こした場合は、共有変数へのアクセスと同様に共有バスへアクセスを出し、共有メモリをアクセスすることでキャッシュをリフィルする。この方式では、並列プログラムの作成にあたり変数がプロセッサ間で共有されているか否かを区別する必要がなくプログラム作成上の制約が少ないことが特徴であるが、局所変数のアクセス性能を高めることができない。

【0005】 密結合マルチプロセッサで高いシステム性能を得るためには、共有バス上のトラフィックを低減し、より多くのプロセッサを結合することが肝要であるが、この方式の共有メモリ型マルチプロセッサでは局所変数のアクセスにも共有バスを使用しなければならず、多数のプロセッサを結合することが事実上困難であるという課題があった。

【0006】 近年、半導体集積回路の製造技術は長足の

進歩を遂げ、今日に至っては汎用計算機の中央処理装置（CPU）のみならず、周辺のキャッシュメモリや記憶管理機構（MMU）をも1個のチップ上に集積することが困難となった。いわゆる、マイクロプロセッサと呼ばれるチップである。主記憶に用いるDRAMにおいては1チップあたり4ないし16メガビットの集積度が実現され、更に高い集積度を目指した研究開発が行なわれている。これらの技術の恩恵を受けて、ワークステーションと呼ばれる小型計算機は、ディスク、LAN回線といった周辺機器の制御装置までを含めて1枚の基板上に集積する。メモリとCPUが密に結合することで高い性能を実現し、CPUの処理性能のみに注目すれば同世代の汎用計算機と肩を並べるまでに至った。この様に今日の技術をもってすれば、実用的なレベルでCPUとメモリを同一基板上に搭載することは可能である。

【0007】図2はメモリを各プロセッサ基板に分散配置した分散共有メモリ型マルチプロセッサの構成図を示す。この構成では、CPUとメモリを同一基板上に搭載し、各基板1010-1～1010-nはバス駆動バッファを経由して共有バス1012で結合されている。この構成で、同一基板上のメモリを局所変数に割り当てれば、局所変数のアクセスは共有バスを経由しなくてもアクセス可能である。そして、プロセッサ間で共有する変数は、共有を行うプロセッサのいずれかの主メモリ上におき、共有バスを用いてアクセスする。

【0008】しかし、既存の並列プログラムは共有変数と局所変数を区別して記述されておらず、新たに作成する並列プログラムにおいても、これを区別して記述することは大きな制約となる。また、同一基板上にCPUとメモリを搭載するにはマイクロプロセッサを用いるが、既存のマイクロプロセッサに共有変数と局所変数を区別してアクセスする手段を持ち合わせていないものが多い。

【0009】メモリシステムの振舞いとしては、あるアクセスが共有変数へのアクセスである可能性が否定できない限り、共有バスを用いて他の基板上のプロセッサのキャッシュメモリにアクセスを行なった旨を伝達しなければ、一貫性を保持することができないため、全てのアクセスを共有バス上に伝達する必要が生じる。

【0010】よって、このような構成を用いても、共有バスのトラフィックは減少せず、システムの拡張性を高められないという問題点があった。

【0011】

【発明が解決しようとする課題】以上まとめると、従来のマルチプロセッサでは、全てのアクセスについて共有バスを経由して共有メモリにアクセスするという動作が行われるため、共有バスのトラフィックが増大し、システムの拡張性を高められないという問題点があった。

【0012】本発明は以上の問題点を解決するためになされたものであり、その目的とするところは、並列プロ

グラムに変数アクセスに対する制約を加えることなく、共有バスの不必要なトラフィックを削減し、システム拡張性の高いマルチプロセッサシステムを提供することにある。

【0013】

【課題を解決するための手段】上記課題を解決するために、本発明は、共有バスを介して結合された複数のプロセッサ・ユニットから成る分散共有メモリ型マルチプロセッサ・システムであって、各プロセッサ・ユニットが、CPUと、内部バスを介してCPUと接続され、システムの共有メモリの分散された一部を記憶する主メモリと、内部バスを介して主メモリと接続され、共有メモリの選択されたデータ・エントリをキャッシュするCPUキャッシュメモリと、内部バスを介して主メモリとCPUキャッシュメモリと接続され、内部バスと共有バスをインタフェースする共有管理部で、該各プロセッサ・ユニット上の主メモリの各データ・エントリが他のプロセッサ・ユニット上のCPUキャッシュメモリに共有されているか否かを示す主メモリの各データ・エントリの共有状態を記憶する主記憶タグ手段と、CPUによる主メモリの或るデータ・エントリに対する書き込み命令の実行に対して、主記憶タグ手段の該或るデータ・エントリの共有状態が或るデータ・エントリは他のプロセッサ・ユニット上のCPUキャッシュメモリに共有されていることを示している時に、該他のプロセッサ・ユニット上のCPUキャッシュメモリの該或るデータ・エントリを無効化するように該他のプロセッサ・ユニットに命令する無効化命令を共有バス上に発行する命令手段と、CPUキャッシュメモリに記憶された選択されたデータ・エントリのアドレスを記憶するキャッシュ・タグ手段と、某データ・エントリに対する無効化命令を共有バスから受け取り該某データ・エントリのアドレスがキャッシュ・タグ手段に記憶されている時に、CPUキャッシュメモリの該某データ・エントリを無効化する制御手段と、を含むものと、から成るシステムを提供する。

【0014】又、本発明は、共有バスを介して結合された複数のプロセッサ・ユニットから成る分散共有メモリ型マルチプロセッサ・システムであって、各プロセッサ・ユニットが、CPUと、内部バスを介してCPUと接続され、システムの共有メモリの分散された一部を記憶する主メモリと、内部バスを介して主メモリと接続され、共有メモリの選択されたデータ・エントリをキャッシュするCPUキャッシュメモリと、内部バスを介して主メモリとCPUキャッシュメモリと接続され、内部バスと共有バスをインタフェースする共有管理部で、CPUキャッシュメモリの各データ・エントリが該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないか否かを示すCPUキャッシュメモリの各データ・エントリのキャッシュ状態を記憶するキャッシュ状態タグ手段と、共有

バスを介して受け取った別のプロセッサ・ユニットからの或るデータ・エントリに対するアクセス要求に対して、キャッシュ状態タグ手段の該或るデータ・エントリのキャッシュ状態が該或るデータ・エントリは該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないことを示している時に、該別のプロセッサ・ユニット上の主メモリの該或るデータ・エントリへのアクセスに介入する介入手段と、介入手段が該別のプロセッサ上の主メモリの該或るデータ・エントリへのアクセスに介入した時に、キャッシュ状態タグ手段の該或るデータ・エントリのキャッシュ状態を、該或るデータ・エントリの所有権を持った一キャッシュ状態から該或るデータ・エントリの所有権を持たない他のキャッシュ状態に状態遷移させ、該各プロセッサ・ユニットからの某データ・エントリへのアクセスが他のプロセッサ・ユニットの介入手段により介入された時に、キャッシュ状態タグ手段の該某データ・エントリのキャッシュ状態を、該某データ・エントリの所有権を持たない一キャッシュ状態から該某データ・エントリの所有権を持った他のキャッシュ状態に状態遷移させるように制御する制御手段と、を含むものと、から成るシステムを提供する。

【0015】又、本発明は、共有バスを介して結合された複数のプロセッサ・ユニットからなる分散共有メモリ型マルチプロセッサ・システムであって、各プロセッサ・ユニットが、CPUと、内部バスを介してCPUと接続され、システムの共有メモリの分散された一部を記憶する主メモリと、内部バスを介して主メモリと接続され、共有メモリの選択されたデータ・エントリをキャッシュするCPUキャッシュメモリと、内部バスを介して主メモリとCPUキャッシュメモリと接続され、内部バスと共有バスをインタフェースする共有管理部で、該各プロセッサ・ユニット上の主メモリの各データ・エントリが他のプロセッサ・ユニット上のCPUキャッシュメモリに共有されているか否かを示す主メモリの各データ・エントリの共有状態を記憶する主記憶タグ手段と、CPUキャッシュメモリの各データ・エントリが該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないか否かを示すCPUキャッシュメモリの各データ・エントリのキャッシュ状態を記憶するキャッシュ状態タグ手段と、CPUによる主メモリの或るデータ・エントリに対する読み出し／書き込み命令の実行に対して、主記憶タグ手段の該或るデータ・エントリの共有状態が該或るデータ・エントリは他のプロセッサ・ユニット上のCPUキャッシュメモリに共有されていることを示している時に、該他のプロセッサ・ユニット上のCPUキャッシュメモリの該或るデータ・エントリを共有バス上に読み出すように該他のプロセッサ・ユニットに命令する読み出し命令を共有バス上に発行する命令手段と、共有バスを介して

受け取った別のプロセッサ・ユニットからの読み出し命令に対して、キャッシュ状態タグ手段の某データ・エントリのキャッシュ状態が該某データ・エントリは該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないことを示している時に、該別のプロセッサ・ユニット上の主メモリの該某データ・エントリへのアクセスに介入する介入手段と、を含むものと、から成るシステムを提供する。

10 【0016】又、本発明は、共有バスを介して結合された複数のプロセッサ・ユニットから成る分散共有メモリ型マルチプロセッサ・システムであって、各プロセッサ・ユニットが、CPUと、内部バスを介してCPUと接続され、システムの共有メモリの分散された一部を記憶する主メモリと、内部バスを介して主メモリと接続され、共有メモリの選択されたデータ・エントリをキャッシュするCPUキャッシュメモリと、内部バスを介して主メモリとCPUキャッシュメモリと接続され、内部バスと共有バスをインタフェースする共有管理部で、CPUキャッシュメモリの各データ・エントリが該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないか否かを示すCPUキャッシュメモリの各データ・エントリのキャッシュ状態を記憶するキャッシュ状態タグ手段と、共有バスを介して受け取った別のプロセッサ・ユニットからの或るデータ・エントリに対するアクセス要求に対して、キャッシュ状態タグ手段の該或るデータ・エントリのキャッシュ状態が該或るデータ・エントリは該各プロセッサ・ユニット上のCPUキャッシュメモリにおいて更新され且つ共有メモリに書き戻されていないことを示している時に、該別のプロセッサ・ユニット上の主メモリの該或るデータ・エントリへのアクセスに介入する介入手段と、を含むものと、から成るシステムを提供する。

【0017】更に、請求項4記載のシステムにおいて、共有管理部が更に、該各プロセッサ・ユニット上の主メモリの各データ・エントリの共有状態を記憶する主記憶タグ手段で、共有状態は該各データ・エントリの所有権が該各プロセッサ・ユニット上に存在することを示す

40 [H] 状態と、該各データ・エントリの所有権が他のプロセッサ・ユニット上に存在することを示す [A] 状態とを含むものと、CPUによる主メモリの某データ・エントリに対するアクセス要求に対して、主記憶タグ手段の該某データ・エントリの共有状態が [A] 状態である時に、該他のプロセッサ・ユニット上のCPUキャッシュメモリの該某データ・エントリを共有バス上に読み出すように該他のプロセッサ・ユニットに命令する読み出し命令を共有バス上に発行する命令手段と、を含むシステムを提供する。

50 【0018】

【作用】請求項1記載の第1の発明によれば、主記憶タグ手段が、自己の主メモリのアドレスが他のプロセッサのキャッシュメモリ上に存在するか（共有か）否か（非共有か）の情報を記憶しているため、CPUが自己の主メモリへの書き込みを行う際に、共有であれば命令手段が共有バス上に無効化命令を出してキャッシュの一貫性を保証し、非共有であれば共有バス上には何も出さずに自己のプロセッサ内部だけで処理を行う。書き込むアドレスをキャッシュメモリ上に持っている他のプロセッサは、キャッシュメモリ上にあるアドレスの情報を待つキャッシュ・タグ手段を用いて、制御手段により前記の無効化命令を共有バスから拾い出して無効化を行う。このように、非共有である場合に共有バスを用いないため、共有バス上のトラフィックを少なくして、キャッシュの一貫性保持が実現できる。

【0019】又、請求項2記載の発明によれば、制御手段がキャッシュ状態タグ手段のキャッシュ状態を、介入先から介入元へ所有権移転が行われるように状態遷移させるので、所有権に併う主メモリへ書き戻す義務が指定されたアドレスのデータ・エントリを主メモリに持たずCPUキャッシュメモリに持つ介入元の基板から指定されたアドレスのデータ・エントリを主メモリに持つ介入先の基板に移転され、後に書き戻しが要求された時には指定されたアドレスのデータ・エントリの書き戻しが主メモリに持つ基板のCPUキャッシュメモリから主メモリへ内部バスのみを使用して行えるようになる。

【0020】又、請求項3記載の発明によれば、主記憶タグ手段が、自己の主メモリのアドレスが他のプロセッサのキャッシュメモリ上に存在するか（共有か）否か

（非共有か）の情報を記憶しているため、CPUが自己の主メモリからの読出し／書き込みを行う際に、共有であれば命令手段が共有バス上に読み出し命令を出してキャッシュの一貫性を保証し、非共有であれば共有バス上には何も出さずに自己のプロセッサ内部だけで処理を行う。読み出すアドレスをキャッシュメモリ上に持っている他のプロセッサは、キャッシュ状態タグ手段が、キャッシュメモリ上にあるアドレスが最新の値でまだ主メモリに書き戻されていない（ダーティ）か否（クリーン）かの情報を記憶しているため、介入手段により前記の読み出し命令を共有バスから拾い出してそのアドレスがダーティであれば介入を行う。このように、非共有である場合に共有バスを用いないため、共有バス上のトラフィックを少なくして、キャッシュの一貫性保持が実現できる。

【0021】又、請求項4及び5記載の発明によれば、主記憶タグ手段の共有状態は[H]状態と[A]状態の2種類のみとなり。メモリ容量やコストの削減が実現可能である上、所有権の所在が共有バス上の介入を待たずに分かるので、共有バス上のトラフィックを低減出来るようになる。

【0022】

【実施例】本実施例の概要を説明すると、CPUとキャッシュメモリと主メモリと共有管理部とが内部バスで結合されたプロセッサ基板を複数共有バスで接続したマルチプロセッサシステムであって、共有管理部は、主メモリのキャッシュラインごとに該キャッシュラインが他の基板上のプロセッサのキャッシュメモリと共有しているか否かを示す共有状態を記憶し、プロセッサが要求したアドレスが他の基板上のプロセッサのキャッシュメモリと共有しているか否かを判定し、共有バス上のトラフィックを監視して、トラフィックが作用するエントリの状態遷移に応じて共有状態を適宜設定し、CPUのキャッシュメモリのエントリごとに該キャッシュラインのCPUのキャッシュメモリ上の状態を記憶し、共有バス上のアクセス要求に対して同一基板上のプロセッサのキャッシュメモリと一貫性保持の動作が必要か否かを判定し、内部バス上のトラフィックを監視して、CPUのキャッシュメモリの状態遷移に応じて、CPUキャッシュのキャッシュ状態を適宜に設定する。

【0023】そして、CPUによって行なわれたアクセスは内部バスに出力され、同一基板上のメモリのアクセスの場合は同一基板上のメモリをアクセスする。

【0024】同時に共有管理部でアドレスを解釈し、該当するアドレスが他の基板上のプロセッサのキャッシュメモリと共有されているか否かを判定する。他のキャッシュメモリと共有が行われていない場合は共有バスを用いず、同一基板上のメモリのアクセスで要求が満たされる。すなわち、この場合の、共有バスを使用することなくアクセス要求を満たすことができ、共有バスのトラフィックを減少させることができる。その間に、共有バスは他の基板の発行する要求の処理を行なうことができる。

【0025】同一基板上のメモリへのアクセスであっても、共有管理部によって他の基板上のプロセッサのキャッシュメモリと共有されていると判定された場合は、内部バスと同時に外部アクセス手段を用いて共有バスにもアクセスを行ない、他の基板上のキャッシュメモリとの一貫性保持を行なう。

【0026】要求されたアドレスが他の基板上のメモリである場合は、共有バスを用いて必要なメモリを持つ基板にアクセスを伝える。要求されたアドレスのメモリを持つ基板上は共有バス上のアドレスを判定してメモリをアクセスし、共有バス上の要求に答える。このとき、メモリを持つ基板上の共有管理部は、そのアドレスが他の基板との共有状態であることを示すように設定する。

【0027】共有バス上にアクセスを行なうと、他の全ての基板の共有管理部が各自の基板上のキャッシュメモリに該当アドレスが存在するか否かを判定する。該当アドレスが存在した場合、共有管理部は該当アドレスが共有であることをアクセス元に応答する。もし、共有バス

上のいずれの基板も共有を応答しなかったならばアクセスを行なった基板の共有管理部は、該当アドレスは非共有になったことを示すように設定する。この場合、それ以降のアクセスは共有バスを用いずに行なうことができるようになる。

【0028】また、共有管理部はCPUのキャッシュメモリの状態を推定しており、共有バス上に要求されたアドレスがキャッシュメモリ上で更新されていた場合や共有バス上の要求が共有変数への書き込みであったなど一貫性保持のための動作が必要と判定した場合のみ、その基板の内部バスを用いてCPUのキャッシュに一貫性保持に必要な動作を指示する。

【0029】内部バス上に発行されたCPUのアクセス要求は以上のような過程を経て完了し、要求されたキャッシュエントリが他のプロセッサのキャッシュと共有しているか否かの応答が示され、これに基づいてCPUのキャッシュはそのエントリの状態を決定する。共有管理部は以上の一連の内部バスのトラフィックを監視し、これに基づいてCPUキャッシュの状態遷移を推定し、CPUのキャッシュの状態変化を設定する。この結果、共有管理部は共有バス上のトラフィックに対して一貫性保持の動作が必要か否かをCPUのキャッシュに問い合わせずに判定する事ができ、CPUを内部バスをその処理に使用し、共有アクセスに伴う不要な干渉を防ぐことができる。

【0030】以下、図面に基いてこの発明に係る分散共有メモリ型マルチプロセッサシステムの実施例について説明する。

【0031】図3は本発明の第1実施例の分散共有メモリ型マルチプロセッサシステムの全体の構成図である。各々のプロセッサ・ユニットはCPU1とキャッシュメモリ2、主メモリ3、共有管理部4を持ち、内部バス5で結合されて、同一基板上に搭載されている。プロセッサ・ユニットを搭載した各基板A、B、C、Dは、バックプレーン上の共有バス6によって結合され、共有管理部4を介して他の基板上のメモリへのアクセスを行なう。

【0032】CPU1は少なくともメモリに対する読み出し命令と書き込み命令を各々1種類以上持ち、プログラムの命令列に従ってプログラムを実行する。

【0033】図4にキャッシュメモリ2の内部構成を示す。キャッシュメモリ2は、各基板上の主メモリの総体から成るマルチプロセッサ全体の共有メモリのデータ・エントリの一部分のコピーを持ち、そのコピーを記憶するデータ記憶部13と、そのコピーの主メモリ上でのアドレスを記憶するアドレスタグ部11と、そのコピーの状態を記憶する状態タグ部12を持つ。本実施例のキャッシュメモリのデータ記憶部は1メガバイトの容量を持つが、キャッシュメモリの容量は本発明の有効性と関係ない。

【0034】アドレスや状態の管理はキャッシュラインと呼ばれるブロックで行う。本実施例では以下の説明でキャッシュラインを64バイトとするが、このキャッシュラインの大きさはCPUが扱うデータ長以上で有れば本発明の有効性を失わない。本発明のキャッシュメモリは16キロラインのキャッシュラインを管理する。

【0035】なお、状態タグ部12に記憶される情報は、他のCPUとの共有状態（共有／非共有）やCPUによって書換が行われ最新の値を保持しているか否かの書換状態（ダーティ／クリーン）、そのキャッシュラインの値が有効であるか否かの有効状態（有効／無効）など、共有メモリの一貫性保持を想定した通常のキャッシュメモリの状態タグと同様の状態情報であり、キャッシュメモリ2は一貫性保持を行う共有メモリのキャッシュメモリとして動作するようキャッシュ制御部17が制御を行う。このようなキャッシュメモリの動作は特開平2-253356号公報および“Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model”, J. Archibald J. Bare, ACM Transaction on Computer Systems, Vol.4 No.4 Nov. 1986などに種々の方法が詳細に紹介されているが、本発明の本質に関わらないのでここでは詳細に説明を行わない。

【0036】本実施例のキャッシュメモリは、コピーバック型のキャッシュ方式を持ち、無効化型の一貫性保持方式をとるものとする。

【0037】主メモリ3はシステムの共有メモリを各基板上に分散したもので、それぞれ異なるアドレスが割り付けてある。すなわち、アドレスによってどの基板上の主メモリをアクセスするかが決定される。本実施例では主メモリの容量を64メガ・バイトとして以下の説明を行うが、主メモリの容量の大きさは本発明の有効性に関係ない。また、主メモリの容量は基板ごとに異なっていて本発明の有効性を損なわない。

【0038】内部バス5はキャッシュミスや一貫性保持動作の際に用いられ、以下のコマンドを持つ。データ転送を行う際は、キャッシュライン単位に行う。

【0039】[CR:] (Coherent Read) 共有読み出し。CPUのread missによって発生し、一貫性を保持しながらデータを読み出す。

40 【0040】[CRI:] (Coherent Read and Invalidate) 排他読み出し。CPUのwrite missによって発生し、データを読み出すとともに他の基板のキャッシュメモリ上の同じアドレスのキャッシュラインを無効化する。

【0041】[CI:] (Coherent Invalidate) 無効化要求。CPUのshared write hitによって発生し、他の基板のキャッシュメモリ上の同じアドレスのキャッシュラインを無効化する。または、他のプロセッサからの無効化要求を共有管理部を介してCPUキャッシュに伝達する場合に発生し、CPUキャッシュの該当エ

50

ントリを無効化する。データ転送を伴わない。

【0042】[WR:] (Normal Write)

書き戻し。キャッシュのリプレースの際にリプレース対象のキャッシュラインが後述するダーティの状態であれば発生し、データを書き込む。

【0043】内部バス5はこれらのコマンドを伝送するコマンド線19、通常のアドレス線20、データ線22の他、読み出したデータが他の基板のキャッシュメモリと共有されているか否かを示す共有指示線18と後述する介入の有無を示す介入指示線21を持つ。この内部バスの共有指示線はその基板の共有管理部が共有していると判断した場合に駆動する。

【0044】共有バス6は一貫性保持動作の際に用いられ、以下のコマンドを持つ。データ転送を行う際は、キャッシュライン単位に行う。

【0045】[RS:] (Read Shared)

共有読み出し。一貫性を保持しながらデータを読み出す。

【0046】[RM:] (Read Modified)

排他読み出し。一貫性を保持しながらデータを読み出し、他の基板のキャッシュメモリの該当エントリを無効化する。

【0047】[INV:] (INValidate)

無効化要求。他の基板上のキャッシュメモリの該当エントリを無効化する。

【0048】[WB:] (Write Back)

書き戻し。プロセッサ基板間に跨る書き戻し。他の基板に読み出され、書き換えられたデータを元の基板の主メモリに書き戻す際に用いる。

【0049】共有バス6は、内部バス5と同様のコマンド線、アドレス線、データ線他に、読み出したデータが他の基板のキャッシュメモリと共有しているか否かを示す共有指示線と後述する介入の有無を示す介入指示線を持つ。この共有バスの共有指示線はそれぞれの基板の共有管理部が指定されたキャッシュラインをその基板のキャッシュメモリが持っていると判断した場合に駆動し、いずれかの基板が駆動すれば共有していると判断される。この様な結線論理はopen collector型出力の論理回路をwired or接続する事で容易に実現できる。

【0050】図5は本実施例の特徴部分である共有管理部4の内部構成図である。共有管理部は、主メモリ3の1ラインに対して1bitの記憶容量を持ち、主メモリの各ラインが他の基板上のプロセッサのキャッシュメモリに存在しているか否かを記憶する主記憶タグメモリ27と、キャッシュメモリ2のタグ部分(アドレスタグ部11、状態タグ部12)と同じ記憶容量を持ち、CPUのキャッシュメモリの状態を推定した結果を記憶し、共有バス上のアクセスが要求したアドレスが同一基板上のプロセッサのキャッシュメモリ上にどのような状態にあるかを記憶するキャッシュアドレスタグメモリ23、キ

ャッシュ状態タグメモリ24とを有する。そして、内部バス5上のアクセスを制御する内部アクセス制御部29と、共有バス6上のアクセスを制御する外部アクセス制御部28と、主記憶タグメモリ27の読み出し・更新を制御する主記憶タグメモリ制御部26と、キャッシュ状態タグメモリ24の読み出し・更新を制御するキャッシュ状態タグメモリ制御部25とが設けられている。

【0051】本実施例の主記憶タグメモリ27は1メガbitの容量を持ち、各ビットは対応する同一基板上の主メモリのキャッシュラインが他のプロセッサのキャッシュメモリに存在している場合(状態G(Global))に値1をとり、同じ基板上のキャッシュメモリにのみコピーがあるかどの基板のキャッシュメモリにもコピーが存在しない場合(状態P(Private))に値0をとる。

【0052】主記憶タグメモリ27の状態は、主記憶タグメモリ制御部26の制御に従い、以下のように遷移する。

【0053】(1)初期状態はP。

【0054】(2)対応するキャッシュラインのエントリに対して他のCPUから共有バスを介してRSまたはRMのアクセス要求があったとき状態Gに遷移する。

【0055】(3)対応するキャッシュラインのエントリに対して共有バスのINVまたはRMを自分のCPUから発行したとき状態Pに遷移する。

【0056】本実施例のキャッシュアドレスタグメモリ23、キャッシュ状態タグメモリ24は、キャッシュメモリのタグ部と同じ、16キロラインのキャッシュラインの情報を管理する。管理する情報は、キャッシュメモリ上のキャッシュラインに対して、各々、主メモリ上でのアドレスと、システム中の状態である。キャッシュラインの状態は以下の5状態のいずれかである。

【0057】なお、以下の説明でダーティとはキャッシュラインが更新されており、かつ更新されたデータを主メモリへまだ書き戻していない状態、つまり主メモリへの書き戻し義務を負っている状態で、クリーンとはそのような義務を負わない状態である。キャッシュラインが共有されている場合、書き戻し義務を負うのはそのうちの1つのキャッシュのみで良いので、ダーティなラインの共有を要求したキャッシュは、共有クリーン状態に設定する。

【0058】[M] 排他ダーティ。他のキャッシュにコピーは存在しない、このラインのデータは変更されており、メモリへ書き戻す必要がある。

【0059】[O] 共有ダーティ。他のキャッシュにコピーが存在する可能性がある。このラインのデータは変更されており、メモリへ書き戻す必要がある。

【0060】[E] 排他クリーン。他のキャッシュにコピーは存在しない。このラインのデータは変更されていないので、メモリへ書き戻す必要はない。

【0061】[S] 共有クリーン。他のキャッシュにコ

ピーが存在する可能性がある。このラインのデータは変更されていないので、メモリへ書き戻す必要はない。

【0062】(以上の4つの状態では、キャッシュラインは有効である。)

【I】無効。このキャッシュラインは有効ではない。

【0063】キャッシュメモリ2の状態タグ部12の状態はキャッシュ制御部17の制御に従い、CPUからの命令、及び、共有管理部が内部バスに発行するコマンドに基づいて、図6の表にまとめた通り以下のように状態を遷移する。

【0064】(1) 初期状態はI。

【0065】(2) 状態Iから自分のCPUが読み出しを行い内部バスの共有指示線が他のCPUと共有している事を示したとき、もしくは、状態Eから他のCPUが共有バスを介して共有読み出しを行ったときに、状態Sに遷移する。

【0066】(3) 状態Iから自分のCPUが読み出しを行い内部バスの共有指示線が他のCPUと共有しない事を示したとき状態Eに遷移する。

【0067】(4) 自分のCPUが書き込みを行ったとき、状態Mに遷移する。

【0068】(5) 状態Mから他のCPUが共有バスを介して共有読み出しを行ったとき状態Oに遷移する。

【0069】(6) 他のCPUが共有バスを介して排他読み出し、または、無効化要求を行ったとき状態Iに遷移する。

【0070】また、状態EまたはSからキャッシュメモリのリプレースが行われて、状態Iに遷移することがある。

【0071】一方、共有管理部4のキャッシュ状態タグメモリ24の状態は、キャッシュ状態タグメモリ制御部25の制御に従い、キャッシュ制御部17が上記のように状態遷移を行うと共に内部バスに発行するコマンド、及び、他の基板の共有管理部が共有バスに発行するコマンドに基づいて、以下のように状態を遷移する。

【0072】まず、自分の基板上の主メモリに対するアクセスの場合は図7の表にまとめた通り以下のように状態を遷移する。

【0073】(1) 初期状態はI。

【0074】(2) 状態Iで主記憶タグがGの場合に、自分のCPUが内部バスを介して共有読み出しを行い共有バスの共有指示線が駆動されているとき、もしくは、状態Eから他のCPUが共有バスを介して共有読み出しを行いキャッシュからの介入が無かったとき、状態Sに遷移する。

【0075】ここで、介入とは、主メモリから値を読み出すとき、最新の値でまだ主メモリに書き戻されていないものがキャッシュメモリに存在する場合に、キャッシュメモリが、バス上の読み出しのコマンドに対して、主メモリからの読み出しを中止するよう介入指示線を駆動

して、自分の持っている最新の値を送る動作のことを言う。この介入は、読み出しが行われる主メモリと最新の値を持っているキャッシュメモリが同じ基板上にあるときには、内部バスを介して行われ、両者が異なる基板上にあるときには、共有バスを介して行われる。

【0076】(3) 状態Iから自分のCPUが内部バスを介して共有読み出しを行い共有バスの共有指示線が駆動されていないとき、状態Eに遷移する。但し、共有管理部の状態タグがSでも、キャッシュメモリの状態タグがIに遷移してしまっている(キャッシュのリプレースにより)ことがあるため、状態Sから同様に状態Eに遷移することがある。

【0077】(4) 自分のCPUが排他読み出しあるいは無効化要求を行い他の基板上のキャッシュを無効化したとき、状態Mに遷移する。

【0078】(5) 状態Mから他のCPUが共有バスを介して共有読み出しを行ったとき、状態Oに遷移する。但し、共有管理部の状態タグがEでも、キャッシュメモリの状態タグがMに遷移してしまっている(書き込みヒットにより)ことがあるため、状態Eから他のCPUが共有バスを介して共有読み出しを行いキャッシュからの介入があったとき(ここで実はMに遷移していたことがわかる)も、状態Oに遷移する。

【0079】(6) 他のCPUが共有バスを介して排他読み出し、または、無効化要求を行ったとき、状態Iに遷移する。リプレースが行われた時も状態Iに遷移する。

【0080】一方、他の基板上の主メモリに対するアクセスの場合は図8の表にまとめた通り以下のように状態を遷移する。

【0081】(1) 初期状態はI。

【0082】(2) 状態Iから自分のCPUが内部バスを介して共有読み出しを行い共有バスの共有指示線が駆動されているとき、もしくは、状態Eから他のCPUが共有バスを介して共有読み出しを行ったとき、状態Sに遷移する。但し、共有管理部の状態タグがEでも、キャッシュメモリの状態タグがIに遷移してしまっている

(キャッシュのリプレースにより)ことがあるため、状態Eから状態Iと同様に状態Sに遷移することがある。

【0083】(3) 状態Iから自分のCPUが内部バスを介して共有読み出しを行い共有バスの共有指示線が駆動されていないとき、状態Eに遷移する。但し、共有管理部の状態タグがSでも、キャッシュメモリの状態タグがIに遷移してしまっている(キャッシュのリプレースにより)ことがあるため、状態Sから状態Iと同様に状態Eに遷移することがある。

【0084】(4) 自分のCPUが排他読み出しあるいは無効化要求を行い他の基板上のキャッシュを無効化したとき、状態Mに遷移する。

【0085】(5) 状態Mから他のCPUが共有バスを

介して共有読み出しを行ったとき、状態Oに移移する。
 【0086】(6)他のCPUが共有バスを介して排他読み出し、または、無効化要求を行ったとき、状態Iに移移する。リブレースが行われた時も状態Iに移移する。

【0087】このように図6～図8にまとめた通り、キャッシュメモリと共有管理部が状態遷移とバス上に発行するコマンドを制御することにより、一部の例外を除いてキャッシュメモリの状態タグ部12と共有管理部のキャッシュ状態タグメモリ24の内部が一致ようになる。一致しない場合も、上記の但し書きで説明したように制御すればエラーは起こらない。さらに共有指示線の駆動に工夫をすることによりエラーを防いでいるが、これについては後述する。

【0088】以下、この第1実施例の共有管理部の動作について説明する。

【0089】図9、10はCPUの読み出し命令に対する本実施例のキャッシュ制御部17及び共有管理部4の動作を示したフローチャートである。CPUが読み出し命令を実行し、キャッシュメモリにミスした場合(S1 No)、内部バス上に共有読み出しCRを発行する(S2)。要求されたアドレスが同一基板上の主メモリのアドレスであれば(S3 Yes)、同一基板上の主メモリが応答する(S4)。同時に共有管理部4の内部アクセス制御部29が内部バス上の要求を解釈し、主記憶タグメモリ27を検索し(S4)、要求された主メモリのキャッシュラインが共有されているか否かを判定する(S5)。共有が行われていなければ、共有管理部は動作を終了し、主メモリのみにアクセスで内部バスの共有読み出しCRは終了する(S6、S7)。これは同一基板上の主メモリ上に割り当てられた局所変数のアクセスに相当し、プログラム実行時のアクセスのほとんどはこの形態のアクセスが行われる。このとき、共有バスを使用せずにアクセスが行われるので、共有バスのトラフィックを軽減できる。また、この間に共有バスは他の基板のアクセス要求を実行できるので他の基板のプロセッサの実行に干渉する可能性が軽減される。内部バスの共有指示線は非共有を示す。

【0090】主記憶タグメモリの検索の結果、要求されたキャッシュラインが他の基板上のCPUキャッシュと共有していると判断された場合は(S5 Yes)、共有管理部4の外部アクセス制御部28を通して共有バス上に共有読み出しRSを発行し、他の基板上のキャッシュメモリと一貫性保持のための動作を行う。他の基板上のキャッシュメモリに最新の値(ダーティなライン)が存在した場合は、その値を持つ基板の共有管理部が介入を行い、最新の値を供給する。その基板のキャッシュメモリにも最新の値が存在しなければ、そのキャッシュラインの値は主メモリ上に存在するので、主メモリから供給された値をキャッシュメモリに供給する(以上S8)。

このとき、主記憶タグメモリの状態は、共有バス上の共有指示線の値に応じて設定し直される(S9)。これにより、前回のアクセスから今回のアクセスの間に共有が解除されていれば検出する事ができる。内部バスの共有指示線は共有バスの共有指示線の状態を伝達する(S10)。

【0091】内部バス上に出力された共有読み出しのアドレスが他の基板上の主メモリのアドレスであれば(S3 No)、共有管理部4は共有バス上に共有読み出しRSを発行し、一貫性保持のための動作を行う(S11)。要求されたアドレスの主メモリを持つ基板はこの共有読み出しRSに回答し、共有管理部の主記憶タグメモリの状態をGに設定し、共有が開始された事を記憶する。内部バスの共有指示線は共有バスの共有指示線が共有/非共有のどちらであっても共有を示す(S12)。

【0092】以上のように読み出されたキャッシュラインはキャッシュメモリに記憶され、キャッシュラインの状態は、内部バスの共有指示線が共有を示している場合は状態Sに(S15)、非共有を示している場合は状態Eに(S14)設定される。共有管理部のキャッシュ状態タグは共有バスの共有指示線の示す値によって状態をSまたはEに設定する。

【0093】ここで注意したいのは、S10とS12の違いである。これが前述したエラーを防ぐための工夫で、他の基板上の主メモリを読み出して共有バスの共有指示線が非共有を示した場合に、共有管理部のキャッシュ状態タグは状態Eに設定されるが、共有管理部は、共有バスの共有指示線が非共有を示しているにも関わらず内部バスの共有指示線を共有にするのでCPUのキャッシュは状態Sに設定される。もし、CPUキャッシュの状態をEと設定してしまうとこのエントリに書き込みが行われてもCPUは無効化要求を出さずに状態Mに移移してしまい、共有管理部のキャッシュ状態タグが書換を認識できず、それ以降の共有バス上の読み出し要求に正しい応答ができなくなってしまう。本実施例では、わざとCPUキャッシュの状態と共有管理部のキャッシュ状態タグの状態を違える事でこの問題を回避する。CPUキャッシュの状態がSであれば、CPUキャッシュは書き込みの際に無効化要求を内部バスに発行するので共有管理部は書き込みを認識する事ができる。この場合、キャッシュ状態タグが状態Eである事から内部バス上の無効化要求を共有バス上に伝達する必要がない事が判り、共有バス上のトラフィックを増加させる事にはならない。

【0094】図11は、CPUの書き込み命令に対する本実施例のキャッシュ制御部17及び共有管理部4の動作を示したフローチャートである。CPUが書き込み命令を実行し、キャッシュメモリにミスした場合(S21 ミス)、内部バス上に排他読み出しCRIを発行する(S32)。要求されたアドレスが同一基板上の主メモ

りのアドレスであれば(S33Yes)、同一基板上の主メモリが応答する。同時に共有管理部の内部アクセス制御部29が内部バス上の要求を解釈し、主記憶タグメモリ27を検索し(S34)、要求された主メモリのキャッシュラインが共有されているか否かを判定する(S36)。共有が行われていなければ、共有管理部は動作を終了し、主メモリのみのアクセスで内部バスの排他読み出しCRIは終了する(S39)。これは同一基板上の主メモリ上に割り当てられた局所変数への書き込みに相当し、プログラム実行時のアクセスのほとんどはこの形態のアクセスが行われる。このとき、共有バスを使用せずにアクセスが行われるので、共有バスのトラフィックを軽減する事ができる。また、この間に共有バスは他の基板のアクセス要求を実行できるので他の基板のプロセッサの実行に干渉する可能性が軽減される。

【0095】主記憶タグメモリの検索の結果、要求されたキャッシュラインが他の基板上のCPUキャッシュと共有していると判断された場合は(S36Yes)、共有管理部の外部アクセス制御部28を通して共有バス上に排他読み出しRMを発行し、他の基板上のキャッシュメモリと一貫性保持のための動作を行う(S37)。他の基板上のキャッシュメモリに最新の値が存在した場合は、その値を持つ基板の共有管理部が介入を行い、最新の値を供給する。どの基板のキャッシュメモリにも最新の値が存在しなければ、そのキャッシュラインの値は主メモリ上に存在するので、主メモリから共有された値をキャッシュメモリに共有する。このとき、他の基板上のキャッシュメモリは無効化が行われ排他読み出しRMを発行したキャッシュメモリのみが排他的にコピーを持つ事になる。よって、主記憶タグメモリは状態Pに設定される(S38)。

【0096】内部バス上に出された排他読み出しCRIのアドレスが他の基板上の主メモリのアドレスであれば(S33No)、共有管理機構は共有バス上に排他読み出しRMを発行し、一貫性保持のための動作を行う(S35)。要求されたアドレスの主メモリを持つ基板はこの排他読み出しRMに回答し、共有管理部の主記憶タグメモリの状態をGに設定し、共有が開始された事を記憶する。

【0097】以上のように読み出されたキャッシュラインはキャッシュメモリに記憶され、CPUの書き込み命令によって更新される。キャッシュラインの状態はMに設定され(S40)、これに応じて共有管理部のキャッシュ状態タグも状態Mに設定される。

【0098】CPUが書き込み命令を実行し、キャッシュメモリにヒットしてもキャッシュラインの状態が共有(状態タグ部12の状態がSまたはO)であれば(S21共有ヒット)、他のキャッシュメモリ上のコピーを無効化する必要があり、内部バス上に無効化要求CIを発行する(S22)。共有管理部の内部アクセス制御部が

内部バス上の要求を解釈し、要求されたアドレスが同一基板上の主メモリのアドレスであれば(S23Yes)、共有管理部の共有バス制御部を通して共有バス上に無効化要求INVを発行し、他の基板上のキャッシュメモリのエントリを無効化する(S26)。そして、主記憶タグメモリを状態Pに設定する(S27)。

【0099】内部バス上に出された無効化要求CIのアドレスが他の基板上の主メモリのアドレスであれば(S23No)、共有管理部はキャッシュ状態タグメモリ24を調べて(S28)状態が共有(SまたはO)であれば(S29Yes)共有バス上に無効化要求INVを発行し、他の基板上のキャッシュメモリのエントリを無効化する(S30)。要求元のエントリは無効化されずに残るため、主メモリをもつ基板上の共有管理部の主記憶タグメモリの状態はGのままに残し、共有が継続している事を記憶する。尚、S28で状態がEであれば(S29No)、内部バス上の無効化要求CIはCPUキャッシュで書き込みヒットが起こったことを示すものである(前述した工夫)から、共有バスにコマンドを伝達することはしない。

【0100】以上のように無効化の行われたキャッシュラインはCPUの書き込み命令によって更新され、キャッシュラインの状態はMに設定される(S31)。これに応じて共有管理部のキャッシュ状態タグも状態Mに設定される。

【0101】さて、全ての基板の共有管理部4の外部アクセス制御部28は共有バス上のトラフィックを監視しており、共有バス上のコマンドと対応するキャッシュ状態タグの状態によって以下のような一貫性保持動作を行う。

【0102】図12は共有バス6上の共有読み出しRSに対する共有管理部4及びキャッシュ制御部17の動作を示すフローチャートである。指定されたアドレスの主メモリを持つ基板の場合は(S41Yes)、主メモリをアクセスするために内部バスに共有読み出しCRを発行する(S42)。この際に、キャッシュメモリの対応する状態タグ部12の状態がIまたはSであればCPUのキャッシュメモリから最新の値を読み出す必要も状態遷移を行う必要もないので、内部バスに共有読み出しCRを発行せず、単に主メモリの読み出しだけを行っても良い。そして、主メモリの読み出しを行うが、キャッシュメモリの状態がMまたはOであれば介入が行われる

(S43)。CPUキャッシュメモリは状態Eで書き込みを行なうと内部バスにコマンドを発行することなく状態Mに遷移するので、共有管理部はこの遷移を認識できない。よって、キャッシュ状態タグの状態がEであっても、CPUのキャッシュメモリの状態はMである可能性が残り、内部バスに発行するコマンドは共有読み出しCRでなければならない。

【0103】共有読み出しCRの結果、CPUキャッシュ

21

メモリの状態タグ部12の状態は、キャッシュ制御部17により

MまたはOならばOに

EまたはSならばSに

IならばIに

それぞれ遷移する(S44)。これに合わせて共有管理部のキャッシュ状態タグメモリ24の状態は、

MまたはO、またはEでキャッシュからの介入が起これば、Oに

S、またはEでキャッシュからの介入が起これば、Sに

IならばIに

それぞれ遷移する(S45)。また、共有管理部の主記憶タグメモリはGに設定される(S46)。

【0104】内部バスの共有指示線はCPUキャッシュメモリが状態Iであれば非共有を、それ以外では共有を示し、この値を共有バスの共有指示線に伝達すると共に、主メモリ又はキャッシュメモリから読み出した値を共有バスに出力する(S47)。

【0105】指定されたアドレスの主メモリを持たない基板の場合は(S41No)、対応するキャッシュ状態タグの状態に応じて内部バスの使用を決定する(S48, S49)。状態がIの場合はこの基板上のキャッシュメモリには有効なコピーが存在しないので一貫性保持のための動作は不要であり、共有指示線の駆動も行わない。状態がI以外の場合は有効なコピーを持つので共有指示線を駆動する(S52, S54)。状態がMまたはOの場合はCPUのキャッシュメモリに最新の値が存在するので内部バスに共有読み出しCRを発行して最新の値をキャッシュメモリから読み出すと共に、共有バスの介入指示線に介入を示し、別の基板上の主メモリに代わって値を共有する(S50, S52)。状態がMまたはO以外の場合は内部バスをアクセスしない。

【0106】CPUキャッシュメモリの状態タグは、MまたはOならばOに(その他はそのまま)共有管理部のキャッシュ状態タグは、MまたはOならばOにそれぞれ遷移する(S51)。

【0107】図13は共有バス上の排他読み出しRMに対する共有管理部4及びキャッシュ制御部17の動作を示すフローチャートである。指定されたアドレスの主メモリを持つ基板の場合は(S61Yes)、主メモリをアクセスするために内部バスに排他読み出しCRIを発行する(S62)。この排他読み出しCRIが行なわれる(S63, S65)に伴い、CPUキャッシュメモリの状態と、共有管理部のキャッシュ状態タグの状態はIに遷移する。また、共有管理部の主記憶タグメモリはGに設定される(S64)。

【0108】指定されたアドレスの主メモリを持たない基板の場合は(S61No)、対応するキャッシュ状態タグの状態に応じて内部バスの使用を決定する(S6

22

6, S67)。状態がIの場合はこの基板上のキャッシュメモリには有効なコピーが存在しないので一貫性保持のための動作は不要であり、内部バスへのコマンド発行は行わない。状態がSまたはEの場合は、無効化要求CIを内部バスに発行する(S71)。状態がMまたはOの場合は、CPUのキャッシュメモリに最新の値が存在するので内部バスに排他読み出しCRIを発行して最新の値をキャッシュメモリから読み出すと共に、共有バスの介入指示線に介入を示し、キャッシュを無効化する

(S68)。読み出した値は主メモリに代わって要求元に供給する(S70)。CPUキャッシュメモリの状態と、共有管理部のキャッシュ状態タグの状態はいずれもIに遷移する(S69, S72)。

【0109】図14は共有バス上の無効化要求INVに対する共有管理部4及びキャッシュ制御部17の動作を示すフローチャートである。指定されたアドレスに対応するキャッシュ状態タグの状態がIでなければ(S81, S82)。内部バスに無効化要求CIを発行し、CPUのキャッシュメモリの有効なエントリを無効化する(S83)。CPUキャッシュメモリの状態と、共有管理部のキャッシュ状態タグの状態はIに遷移し、要求されたアドレスを割り当てられた主メモリを持つプロセッサ基板の場合は主記憶タグの状態はGに設定される(S84)。

【0110】共有バス上の書き戻しWBに対しては、主メモリを持つ基板のみが応答し、送られてきた値を取り込み主メモリに書き戻す。他の基板は何も応答せずに動作を終了する。

【0111】以下、例題を用いて本実施例の動作の例を示す。図15は例題の構成図である。ここでアドレスの先頭の0xは16進数であることを示す。

【0112】プロセッサ基板PB₀はアドレス0x0000 0000から0x03ff ffffの主メモリM₀を持ち、プロセッサ基板PB₁はアドレス0x0400 0000から0x07ff ffffの主メモリM₁を持ち、プロセッサ基板PB₂はアドレス0x0800 0000から0x0bff ffffの主メモリM₂を持ち、プロセッサ基板PB₃はアドレス0x0c00 0000から0x0fff ffffの主メモリM₃を持つ。

【0113】アドレス0x0000 1000のキャッシュラインはプロセッサ基板PB₀の主メモリM₀にあるが、どのプロセッサのキャッシュメモリにも存在しない。よって、対応する主記憶タグメモリMTag₀の状態はPである。

【0114】アドレス0x0400 2000のキャッシュラインはプロセッサ基板PB₁の主メモリM₁にあるが、プロセッサ基板PB₁のキャッシュメモリC₁に状態Eで存在する。よって、対応する主記憶タグメモリMTag₀の状態はPである。

【0115】図16はプロセッサ基板PB₀のCPU(CPU₀)がアドレス0x0000 1000を読み出し(1)

たときの動作を示している。プロセッサ基板PB₀のキャッシュC₀は読み出しミスを起こし(2)、プロセッサ基板PB₀の内部バスB₀に共有読み出しCRを発行する(3)。プロセッサ基板PB₀の主記憶タグメモリMTag₀は状態Pであるので、共有バスSBへのアクセスは行わず(5)、プロセッサ基板PB₀の主メモリM₀を読み出して(6)キャッシュC₀に記憶し、CPU(CPU₀)に値を供給する(8)。

【0116】プロセッサ基板PB₀のキャッシュメモリC₀とキャッシュ状態タグメモリCTag₀は状態Eに移移する(4)(7)。以降、プロセッサ基板PB₀のCPU(CPU₀)からの読み出しはキャッシュC₀にヒットし、内部バスB₀に要求がでない。

【0117】図17はこの状態からプロセッサ基板PB₀のCPU(CPU₀)がアドレス0x0000 1000へ書き込みを行った(1)ときの動作を示している。プロセッサ基板PB₀のキャッシュC₀は書き込みヒットを起こし(2)、内部バスB₀にアクセスすることなく、状態をMに移移する(3)。以降、キャッシュC₀から追い出されるまで、プロセッサ基板PB₀のCPU(CPU₀)からの読み出しと書き込みのいずれのアクセスもキャッシュC₀にヒットし、内部バスB₀に要求がでない(4)。この様に、同一基板上の局所変数は共有バスSBを用いることなくアクセスすることができる。共有管理部S₀のキャッシュ状態タグCTag₀は状態Eのままに残るが、共有読み出しを行なう際は主メモリM₀にアクセスする関係で共有読み出しを行なうので問題を生じない。他のプロセッサ基板から共有バスSBを介して共有読み出しを行なえば不一致の状態が解消される。

【0118】図18はプロセッサ基板PB₀のCPU(CPU₀)がアドレス0x0400 2000を読み出し(1)たときの動作を示している。プロセッサ基板PB₀のキャッシュC₀は読み出しミスを起こし(2)、プロセッサ基板PB₀の内部バスB₀に共有読み出しCRを発行する(3)。このアドレスはプロセッサ基板PB₀の主メモリM₀ではないので共有バスSB上に共有読み出しRSを発行する(4)。

【0119】プロセッサ基板PB₁からPB₂は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板PB₁からPB₂はキャッシュ状態タグCTag₁、CTag₂が無効状態Iなので動作を終了する(5)。プロセッサ基板PB₁は要求されたアドレスの主メモリM₁を持ち、キャッシュ状態タグCTag₁が状態Eなのでその内部バスB₁に共有読み出しCRを発行し(6)、主メモリM₁がキャッシュラインの値を供給する(7)。このときプロセッサ基板PB₁のCPUキャッシュC₁は共有が開始されたことを認識し、状態EからSに移移する(8)。また、プロセッサ基板PB₁のキャッシュ状態タグCTag₁も状態Sに、主記憶タグMTag₁は状態Gに移移する(8)。

【0120】読み出されたキャッシュラインは共有バスSBを介してプロセッサ基板PB₀に供給される

(9)。このとき共有バスSBの共有指示線は共有を指示している。プロセッサ基板PB₀のキャッシュ状態タグCTag₀は状態Sに設定し(10)、内部バスB₀に共有指示線に共有を指示しながらCPUキャッシュC₀にキャッシュラインの値を供給する(11)。CPUキャッシュC₀は状態Sとなり(12)、このCPUキャッシュC₀からCPU(CPU₀)にキャッシュラインの値が読出される(13)。

【0121】図19は、この状態からプロセッサ基板PB₀のCPU(CPU₀)がアドレス0x0400 2000に書き込みを行った(1)ときの動作を示している。プロセッサ基板PB₀のキャッシュC₀は共有ヒットを起こし(2)、内部バスに無効化要求CIを発行する(3)。プロセッサ基板PB₀の共有管理部S₀は同一基板上の主メモリM₀のアドレスでないので、共有バスSBに無効化要求INVを発行する(4)。

【0122】プロセッサ基板PB₁からPB₂は共有バスSB上の無効化要求INVを見て動作を開始するが、プロセッサ基板PB₁からPB₂はキャッシュ状態タグCTag₁、CTag₂が無効状態Iなので動作を終了する(5)。プロセッサ基板PB₁はキャッシュ状態タグCTag₁が状態Sなので無効化要求CIを内部バスB₁に伝達する(6)。プロセッサ基板PB₁のCPUキャッシュC₁はこの無効化要求CIによって状態をIに移移し(7)、キャッシュ状態タグCTag₁も状態Iに移移する(8)が、主記憶タグMTag₁の状態はGのままである。プロセッサ基板PB₁の無効化が終了するとプロセッサ基板PB₀のキャッシュ状態タグCTag₀は状態Mに移移し(9)、CPUキャッシュC₀の状態もMに移移して(10)キャッシュC₀に対する書き込みが行われる(11)。

【0123】図20は、この状態からプロセッサ基板PB₁のCPU(CPU₁)がアドレス0x0400 2000に読み出しを行った(1)ときの動作を示している。プロセッサ基板PB₁のキャッシュC₁は読み出しミスを起こし(2)、内部バスB₁に共有読み出しCRを発行する(3)。プロセッサ基板PB₁の共有管理部S₁は同一基板上の主メモリM₁のアドレスではあるが、主記憶タグMTag₁の状態がGであるので主メモリM₁のアクセス(4)と同時に共有バスSBにも共有読み出しRSを発行する(4)。

【0124】プロセッサ基板PB₀とPB₁からPB₂は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板PB₁からPB₂はキャッシュ状態タグCTag₁、CTag₂が無効状態Iなので動作を終了する(5)。プロセッサ基板PB₀はキャッシュ状態タグCTag₀が状態Mなので要求に介入する(6)。この時点でプロセッサ基板PB₁の主メモリM₁のアク

25

セスは中断される(7)。プロセッサ基板PB₀の共有管理部S₀は内部バスB₀に共有読み出しCRを発行し(8)、CPUキャッシュC₀からキャッシュラインの値を取り出す。この共有読み出しCRによってCPUキャッシュC₀は状態をOに遷移する(9)。プロセッサ基板PB₀の共有管理部S₀は読み出されたキャッシュラインの値を共有バスSBにのせると共にキャッシュ状態タグCTag₀を状態Oに遷移する(10)。

【0125】プロセッサ基板PB₁の共有管理部S₁はえられた値を内部バスB₁に返すと共に、キャッシュ状態タグCTag₁の状態をSに遷移する(11)。プロセッサ基板PB₁のCPUキャッシュC₁は、主メモリM₁の古い値ではなく、共有管理部S₁が供給するプロセッサ基板PB₀で更新された新しいキャッシュラインの値をキャッシュC₁に取り込んで状態をSに設定し(12)このCPUキャッシュC₁からCPU(CPU₁)にキャッシュラインの値が読出される(13)。以上の動作によってプロセッサ基板PB₁の読み出しはプロセッサ基板PB₀で行われた書き込みの結果が反映され、キャッシュ間の一貫性が保たれる。

【0126】図21は、この状態からプロセッサ基板PB₂のCPU(CPU₂)がアドレス0x0400 2000に読み出しを行った(1)ときの動作を示している。プロセッサ基板PB₂のキャッシュC₂は読み出しミスを起こし(2)、内部バスB₂に共有読み出しCRを発行する(3)。プロセッサ基板PB₂の共有管理部S₂は同一基板の主メモリM₂のアドレスでないので、共有バスSBに共有読み出しRSを発行する(4)。

【0127】プロセッサ基板PB₀からPB₁とPB₂は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板PB₂はキャッシュ状態タグCTag₂が無効状態Iなので動作を終了する(5)。プロセッサ基板PB₁は要求されたアドレスの主メモリM₁を持つので、主メモリM₁のアクセスを行うために内部バスB₁に共有読み出しCRを発行する(6)。プロセッサ基板PB₀はキャッシュ状態タグCTag₀が状態Oなので要求に介入する(7)。この時点でプロセッサ基板PB₁の主メモリM₁のアクセスは中断される

(8)。プロセッサ基板PB₀の共有管理部S₀は内部バスB₀に共有読み出しCRを発行し、CPUキャッシュC₀からキャッシュラインの値を取り出す(8)。プロセッサ基板PB₀の共有管理部S₀は読み出されたキャッシュラインの値を共有バスSBに載せる。

【0128】プロセッサ基板PB₂の共有管理部S₂は得られた値を内部バスB₂に返すと共に、キャッシュ状態タグCTag₂の状態をSに遷移する(9)。プロセッサ基板PB₂のCPUキャッシュC₂は、内部バスB₂に返された値を取り込んで状態をSに設定する(10)。この値はプロセッサ基板PB₁の主メモリM₁の古い値ではなく、プロセッサ基板PB₀で更新された新

26

しい値であるので、プロセッサ基板PB₀で行われた書き込みの結果が反映され、共有しているプロセッサ基板PB₀からPB₂の間でキャッシュ間の一貫性が保たれる。

【0129】図22はプロセッサ基板PB₀のCPU(CPU₀)が、アドレス0x0000 1000に対して、プロセッサ基板PB₁上のキャッシュメモリC₁がこのアドレスのキャッシュラインを状態Oで持ち、プロセッサ基板PB₂上のキャッシュメモリC₂がこのアドレスのキャッシュラインを状態Sで持ち、プロセッサ基板PB₀上の主記憶タグMTag₀がこのアドレスのキャッシュラインの共有状態が状態Gであることを示している時に、書き込みを行った(1)ときの動作を示している。プロセッサ基板PB₀のキャッシュメモリC₀は書き込みミスを起こし(2)、内部バスB₀に排他読み出しCRIを発行する(3)。このアドレスはプロセッサ基板PB₀上の主メモリM₀に割当てられたものであるが、このプロセッサ基板PB₀上の主記憶タグMTag₀の示す共有状態が状態Gであるので、このプロセッサ基板PB₀の共有管理部S₀は主メモリM₀にアクセスする(4)と共に、共有バスSBに排他読出しRMを発行する(4)。

【0130】プロセッサ基板PB₁からPB₂は共有バスSB上の排他読み出しRMを見て動作を開始するが、プロセッサ基板PB₂はキャッシュ状態タグCTag₂が無効状態Iなので動作を終了する(5)。プロセッサ基板PB₁はキャッシュ状態タグCTag₁が状態Oなのでその内部バスB₁に排他読み出しCRIを発行すると共に、共有バスSBからの要求に介入し(6)、プロセッサ基板PB₀上の主メモリM₀のアクセスを中断する(7)と共に、プロセッサ基板PB₁上のキャッシュメモリC₁からこのアドレスのキャッシュラインが共有バスSBに読み出される。プロセッサ基板PB₂はキャッシュ状態タグCTag₂が状態Sなのでその内部バスB₂に無効化要求CIを発行する(6)。

【0131】これにより、プロセッサ基板PB₁のキャッシュメモリC₁は状態をIに遷移し(7)、プロセッサ基板PB₂のキャッシュメモリC₂も状態をIに遷移して(7)、当該アドレスのキャッシュラインを無効化する。これに伴い、プロセッサ基板PB₁のキャッシュ状態タグCTag₁も状態Iに遷移し(8)、プロセッサ基板PB₂のキャッシュ状態タグCTag₂も状態Iに遷移する(8)。

【0132】プロセッサ基板PB₁及びPB₂の無効化が終了するとプロセッサ基板PB₀のキャッシュ状態タグCTag₀が状態Mに遷移し、主記憶タグMTag₀が状態Pに遷移する(9)。そして、キャッシュメモリC₀の状態が状態Mに遷移し(10)、キャッシュメモリC₀に対して書き込みが行われる(11)。

【0133】次に、本発明の第2実施例の分散共有メモ

リ型マルチプロセッサシステムについて説明する。

【0134】この第2実施例は上記第1実施例において異なるプロセッサ基板の間での所有権の移転に関する改良を加えたもので、ここで所有権とは最新の値をキャッシュから主メモリに書き戻す義務があることを意味する。

【0135】本実施例においては、共有バス6は更に共有バス6上のメモリアクセス要求の要求元であるプロセッサ基板を同定する要求元識別線を備えている。

【0136】あるプロセッサ基板上のCPU1が同基板上の主メモリに割り当てられたアドレスに対する読み出し命令を実行し、キャッシュメモリ2で読み出しミスが起こった際に、別のプロセッサ基板上のキャッシュメモリ2がこのアドレスの最新の値を持っていた場合、この別のプロセッサ基板は上記第1実施例の場合と同様に介入を行う。

【0137】上記第1実施例では、このような場合でも所有権は介入元である別のプロセッサ基板の方に残すため、後に書き戻しが要求されると、この別のプロセッサ基板が前記あるプロセッサ基板の主メモリに対して共有バス6を介して書き戻しを行わねばならない。

【0138】これに対して、この第2実施例では、介入時に最新の値と共に所有権もこの別のプロセッサ基板から前記あるプロセッサ基板に移転させて、後の書き戻しがこのあるプロセッサ基板内だけで、共有バス6を使わずに行えるようにしている。

【0139】このような所有権の移転を行うために、この第2実施例では、共有バス6上のメモリ・アクセス要求は要求元識別線上の要求元プロセッサ基板を示す情報を併用するようにする。この情報は通常のバスが持つ基板識別番号でも良いが、実質的には要求元のプロセッサ基板が指定されたアドレスが割り当てられた主メモリを持ったプロセッサ基板であるかどうかを示すもので充分であるので、このことを示す1ビットの信号で与えても良い。

【0140】ここで、指定されたアドレスが割り当てられた主メモリを持ち、介入を受けた要求元のプロセッサ基板では、キャッシュメモリ2が最新の値と併せて所有権をも共有バス6を介して受けとるので、キャッシュメモリ2のキャッシュ状態は状態Oにされ、介入を行った別のプロセッサ基板ではキャッシュのキャッシュ状態は状態Sにされる。

【0141】従って、キャッシュメモリ2のキャッシュ状態タグ部12の状態遷移は上記図6にまとめたものに代わって図23にまとめたものとなり、一方共有管理部4のキャッシュ状態タグメモリ24の状態遷移は自分のプロセッサ基板上の主メモリのアクセスの際には上記図7にまとめたものに代わって図24にまとめたものとなり、他のプロセッサ基板上の主メモリへのアクセスの際には上記図8にまとめたものに代わって図25にまとめ

たものとなる。

【0142】又、この第2実施例では、CPUの読み出し命令に対するキャッシュ制御部17及び共有管理部4の動作を示す上記図9、10のフローチャートを、図26、27のように変更する。

【0143】即ち、S8における一貫性保持動作の終了後、この一貫性保持動作において介入があったか否かを決定し(S16)、なかった場合は図9、10の場合と同様にS9に進むが、あった場合は内部バス5の介入指示線を駆動し(S17)、CPUキャッシュの状態を状態Oに設定する(S18)ようにする。これ以外は図9、10と同様で良い。

【0144】更に、共有バス6上の共有読み出しRSに対する共有管理部4とキャッシュ制御部17の動作を示す上記図12のフローチャートを図28の様に変更する。

【0145】即ち、S50の後で、要求されたアドレスの割り当てられた主メモリを持つプロセッサ基板からのアクセスか否かを決定し(S55)、そうでなければ図12の場合と同様にS51に進むが、そうであった場合にはCPUキャッシュメモリの状態タグは、MまたはOならばSに(その他はそのまま)、共有管理部のキャッシュ状態タグは、MまたはOならばSに、それぞれ遷移させ(S56)、そしてS52に進むようにする。これ以外は図12と同様で良い。

【0146】図29はプロセッサ基板PB₀のCPU(CPU₀)が、プロセッサ基板PB₁上のキャッシュメモリC₁で更新されているアドレス0x0000 0000に対して読み出しを行った(1)ときの動作を示している。プロセッサ基板PB₀のキャッシュメモリC₀は読み出しミスを起こし(2)、内部バスB₀に共有読み出しCRを発行する(3)。このアドレスはプロセッサ基板PB₀上の主メモリM₀に割り当てられたものであるが、このプロセッサ基板PB₀上の主記憶タグMTag₀の示す共有状態が状態Gであるので、このプロセッサ基板PB₀の共有管理部S₀は主メモリM₀にアクセスする(4)と共に、共有バスSBに共有読み出しRSを発行する(4)。

【0147】プロセッサ基板PB₁からPB₀は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板PB₁とPB₀はキャッシュ状態タグCTag₁、CTag₀が無効状態Iなので動作を終了する(5)。プロセッサ基板PB₁はキャッシュ状態タグCTag₁が状態Mなので、その内部バスB₁に共有読み出しCRを発行すると共に共有バスSBからの要求に介入し(6)、プロセッサ基板PB₁上のキャッシュメモリC₁からこのアドレスのキャッシュラインを読み出す(7)。この時、このキャッシュメモリC₁のキャッシュ状態がMからSに遷移する。

【0148】これにより、プロセッサ基板PB₀上の主

メモリM₀のアクセスが中断される(7)と共に、キャッシュメモリC₁から読み出した最新の値が共有バスSBを介してプロセッサ基板PB₀に転送される(8)。この時点で、キャッシュ状態タグメモリCTag₀のキャッシュ状態がMからSに移移する(8)。

【0149】次に、プロセッサ基板PB₀においては、キャッシュ状態タグメモリCTag₀のキャッシュ状態がIからOに移移して(9)、所有権が移転されてきたことを示す。共有バスSBを介して転送されたデータはキャッシュメモリC₀にキャッシュされて(10)、キャッシュメモリC₀のキャッシュ状態もIからOに移移して、所有権が移転されてきたことを示す。このキャッシュC₀からCPU(CPU₀)に最新の値が読出される(11)。

【0150】次に、本発明の第3実施例の分散共有メモリ型マルチプロセッサシステムについて説明する。

【0151】この第3実施例は上記第1実施例においてキャッシュメモリ2のキャッシュ状態タグ部12と共有管理部4のキャッシュ状態タグメモリ24で用いるキャッシュ状態をM、E、S、Iの4状態に限定し、状態Oを省略するという改良を加えたものである。

【0152】実際の回路では各キャッシュ状態は2進数のビットで表現されるためキャッシュ状態数を2の中数とすることが望ましく、このためこの第3実施例のようにキャッシュ状態を4状態に限定することによりキャッシュ状態タグ部12やキャッシュ状態タグメモリ24の実現が容易となる。

【0153】4状態だけを使ったキャッシュメモリの一貫性保持方式としてはBerkeley方式とIllinois方式の2方式が知られているが、本実施例ではIllinois方式を採用したものとして説明する。但し、Berkeley方式を用いても同様に実現が可能である。

【0154】Illinois方式では主メモリ3にいわゆるリフレクティブ機能を持たせる。これは、主メモリ3の共有読み出しの際に介入が発生した時に、主メモリ3の読み出し動作を中断する代わりに、介入によって内部バスに転送されてきたデータを主メモリ3に書き込む機能である。この機能によれば、介入を行ったキャッシュは介入と同時に書き戻し義務を果すことになるので、所有権の存在を示す状態O(共有ダーティ)は不要となる。

【0155】従って、共有管理部4のキャッシュ状態タグメモリ24の状態遷移は自分のプロセッサ基板上の主メモリへのアクセスの際には上記図7にまとめたものに代わって図30にまとめたものとなり、他のプロセッサ基板上の主メモリへのアクセスの際には上記図8にまとめたものに代わって図31にまとめたものとなる。

【0156】図32はプロセッサ基板PB₁のCPU(CPU₁)が、プロセッサ基板PB₀上のキャッシュメモリC₀で更新されているアドレス0x0000 1000に対して読み出しを行った(1)ときの動作を示している。

プロセッサ基板PB₁のキャッシュメモリC₁は読み出しミスを起こし(2)、内部バスB₁に共有読み出しCRを発行する(3)。このアドレスはプロセッサ基板PB₁上の主メモリM₁に割り当てられたものではないので、プロセッサ基板PB₁の共有管理部S₁は共有バスSBに共有読み出しRSを発行する(4)。

【0157】プロセッサ基板PB₀とPB₁からPB₁は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板PB₀とPB₁はキャッシュ状態タグCTag₀、CTag₁が無効状態Iなので動作を終了する(5)。プロセッサ基板PB₀はキャッシュ状態タグCTag₀が状態Mなので、その内部バスB₀に共有読み出しCRを発行して、キャッシュメモリC₀からこのアドレスのキャッシュラインを読み出す(6)と共に、内部バスB₀の介入指示線を駆動して(6)このメモリアクセスに介入する。この時、このキャッシュメモリC₀はキャッシュ状態をMからSに移移する。

【0158】これによりプロセッサ基板PB₀上の主メモリM₀のアクセスが中断される(7)と共に、キャッシュメモリC₀の最新の値が主メモリM₀に書き込まれて書き戻し義務が果され、最新の値は共有バスSBを介して、プロセッサ基板PB₁に転送される。この時点で、キャッシュ状態タグメモリCTag₀はキャッシュ状態をMからSに移移し、主記憶タグメモリMTag₀は共有状態をPからGに移移する(7)。

【0159】次に、プロセッサ基板PB₁においては、キャッシュ状態タグメモリCTag₁がキャッシュ状態をIからSに移移し(8)、共有バスSBを介して転送されたデータがキャッシュメモリC₁にキャッシュされて、キャッシュメモリC₁はキャッシュ状態をIからSに移移する(9)。このキャッシュC₁からCPU(CPU₁)に最新の値が読み出される。

【0160】図33はプロセッサ基板PB₁のCPU(CPU₁)が、プロセッサ基板PB₀上のキャッシュメモリC₀で更新されているアドレス0x0000 1000に対して書き込みを行った(1)ときの動作を示している。プロセッサ基板PB₁のキャッシュメモリC₁は書き込みミスを起こし(2)、内部バスB₁に排他読み出しCRIを発行する(3)。このアドレスはプロセッサ基板PB₁上の主メモリM₁に割り当てられたものではないので、プロセッサ基板PB₁の共有管理部S₁は共有バスSBに排他読み出しRMを発行する(4)。

【0161】プロセッサ基板PB₀とPB₁からPB₁は共有バスSB上の排他読み出しRMを見て動作を開始するが、プロセッサ基板PB₀とPB₁はキャッシュ状態タグCTag₀、CTag₁が無効状態Iなので動作を終了する(5)。プロセッサ基板PB₀はキャッシュ状態タグCTag₀が状態Mなので、その内部バスB₀に排他読み出しCRIを発行して、キャッシュメモリC₀からこのアドレスのキャッシュラインを読み出す(6)と共に

に、内部バスB₀の介入指示線を駆動して(6)、このメモリアクセスに介入する。この時、キャッシュメモリC₀はキャッシュ状態をMからIに移す。

【0162】これにより、プロセッサ基板P_{B0}上の主メモリM₀のアクセスが中断される(7)と共に、最新の値は共有バスSBを介してプロセッサ基板P_{B1}に転送される。ここで最新の値はいつれプロセッサ基板P_{B1}で書き替えられるので、主メモリM₀に書き込む必要はない。この時点で、キャッシュ状態タグメモリCTag₀はキャッシュ状態をMからIに移し、主記憶タグメモリMTag₀は共有状態をPからGに移す(7)。

【0163】次に、プロセッサ基板P_{B1}においては、キャッシュ状態タグメモリCTag₁がキャッシュ状態をIからMに移し(8)、共有バスSBを介して転送されたデータがキャッシュメモリC₁にキャッシュされて、キャッシュメモリC₁はキャッシュ状態をIからMに移す(9)。このキャッシュC₁に対してCPU(CPU₁)が書き込みを行う。

【0164】図34はプロセッサ基板P_{B2}のCPU(CPU₂)が、プロセッサ基板P_{B1}上のキャッシュメモリC₁で更新させているアドレス0x0000 1000に対して読み出しを行った(1)ときの動作を示している。プロセッサ基板P_{B2}のキャッシュメモリC₂は読み出しミスを起こし(2)、内部バスB₂に共有読み出しCRを発行する(3)。このアドレスはプロセッサ基板P_{B2}上の主メモリM₂に割り当てられたものではないので、プロセッサ基板P_{B2}の共有管理部S₂は共有バスSBに共有読み出しRSを発行する(4)。

【0165】プロセッサ基板P_{B0}からP_{B1}とP_{B2}は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板P_{B2}はこのキャッシュラインのコピーをキャッシュメモリC₂にも主メモリM₂にも持たないので動作を終了する(5)。プロセッサ基板P_{B1}はキャッシュ状態タグCTag₁が状態Mなので、その内部バスに共有読み出しCRを発行して(5)、キャッシュメモリC₁からこのアドレスの最新の値を読み出す(6)と共に、内部バスの介入指示線を駆動して介入を示す。この時点で、キャッシュメモリC₁はキャッシュ状態をMからSに移させ、共有管理部S₁のキャッシュ状態タグCTag₁もキャッシュ状態をMからSに移させる(7)。そして、最新の値は共有バスSBを介して介入指示線の介入指示と共にプロセッサ基板P_{B2}に転送される(7)。

【0166】プロセッサ基板P_{B0}はキャッシュ状態タグCTag₀が状態Iなので、共有バスSB上の共有読み出しRSには応答しない。

【0167】次に、プロセッサ基板P_{B2}においては、キャッシュ状態タグメモリCTag₂がキャッシュ状態をIからSに移す(8)が、プロセッサ基板P_{B0}のキャッシュ状態タグメモリCTag₀のキャッシュ状態は

Iのままである(8)。

【0168】そして、共有バスSBを介して転送されたデータがキャッシュメモリC₂にキャッシュされて、キャッシュメモリC₂はキャッシュ状態をIからSに移す(9)。このキャッシュC₂からCPU(CPU₂)に最新の値が読み出される。

【0169】一方、プロセッサ基板P_{B0}では、共有バスSBの介入指示線の介入指示に応じて共有バスを介して転送されたこのキャッシュラインの最新の値が主メモリM₀に書き込まれて(9)、書き戻し義務が果たされる。

【0170】次に、本発明の第4実施例の分散共有メモリ型マルチプロセッサシステムについて説明する。

【0171】この第4実施例は上記第1実施例において共有管理部4の主記憶タグメモリ27で用いる共有状態に他のプロセッサ基板上のキャッシュメモリで書き換えられているか否か、即ち他のプロセッサ基板上に所有権が存在するか否か、を示す情報を、他のプロセッサ基板上のキャッシュメモリからのアクセスの有無に加えるという改良を加えたものである。

【0172】この追加の情報は所有権がプロセッサ基板上に不在であるか否かを示すものであるため、主メモリ3に対して読み出しを行う際に介入が起こるか否かを前もって知らせることが可能である。

【0173】この追加の情報を表すビットは共有バス6上に排他読み出しRMと無効化要求INVが発行された時に駆動され、共有バス6上に書き戻しWBが発行された時に解除されることで、所有権の所在を正確に反映出来るようにされる。

【0174】従って、この第4実施例では、主記憶タグメモリ27は以下の3状態P、G、Aのいずれかを取り得る。

【0175】[P]:他のプロセッサ基板的キャッシュにコピーがない。

【0176】[G]:他のプロセッサ基板的キャッシュにコピーが存在する可能性があるが、所有権はこのプロセッサ基板にある。

【0177】[A]:他のプロセッサ基板的キャッシュに所有権がある。

【0178】ここで、所有権が存在するということは、キャッシュにコピーがあることをも意味するので、これら3状態で全ての状況をカバーできる。

【0179】この第4実施例では、主メモリ3に対して読み出しを行なう際に介入が起こるか否かを、共有バス6からの介入を待つことなく主記憶タグメモリの共有状態から前もって知ることが可能なため、指定されたアドレスを割り当てられた主メモリ3を持つプロセッサ基板での動作を主記憶タグメモリの共有状態に応じて以下の様にすることで、共有バス6上のトラフィックを減らすことが可能である。

【0180】[P]:読み書き共に共有バス6を介さず内部バス5のみで行う。

【0181】[G]:読み出しは内部バス5のみで行い、書き込みは共有バス6に無効化要求INVを発行して行う。

【0182】[A]:読み出しは主メモリ3へのアクセスを中断して共有バス6からの介入を待って行い、書き込みは共有バス6に無効化要求INVを発行して行う。

【0183】又、この第4実施例では、キャッシュリファイルを行なった際に共有状態が[P]であれば、他のキャッシュにコピーが存在しないことを意味するので、キャッシュ状態を状態E(排他クリーン)に設定出来る。キャッシュ状態がEの時には書き込み、読み出し、リプレースの全てを内部バス5のみを使って行うことが可能なので、共有バス6上の不要なトラフィックを削減出来る。

【0184】又、自分のプロセッサ基板上の主メモリに対するアクセスであっても、共有状態が[A]であれば、他のプロセッサ基板上のキャッシュが介入を行うので、主メモリ3にアクセスする必要はなく、内部バス5のトラフィックを削減出来る。

【0185】従って、共有管理部4のキャッシュ状態タグメモリ24の状態遷移は自分のプロセッサ基板上の主メモリへのアクセスの際には上記図7にまとめたものに代わって図35にまとめたものとなり、他のプロセッサ基板上の主メモリへのアクセスの際には上記図8にまとめたものに代わって図36にまとめたものとなる。

【0186】図37はプロセッサ基板PB₁のCPU(CPU₁)が、プロセッサ基板PB₀上の主メモリM₀に記憶されているアドレス0x00001000に対して書き込みを行った(1)ときの動作を示している。プロセッサ基板PB₁のキャッシュメモリC₁は読み出しミスを起こし(2)、内部バスB₁に排他読み出しCRIを発行する(3)。このアドレスはプロセッサ基板PB₁上の主メモリM₁に割り当てられたものではないので、プロセッサ基板PB₁の共有管理部S₁は共有バスSBに排他読み出しRMを発行する(4)。

【0187】プロセッサ基板PB₀とPB₁からPB₁は共有バスSB上の排他読み出しRMを見て動作を開始するが、プロセッサ基板PB₁とPB₁は、このキャッシュラインのコピーをキャッシュメモリC₁、C₁にも主メモリM₁、M₁にも持たないので動作を終了する(5)。プロセッサ基板PB₀はこのアドレスが割り当てられた主メモリM₀を持つので、その内部バスB₀に排他読み出しCRIを発行して(5)、主メモリM₀からこのキャッシュラインを読み出す(6)。

【0188】読み出されたデータは共有バスSBを介してプロセッサ基板PB₁に転送される(7)。この時点で主記憶タグメモリMTag₀の共有状態はPからAに移転して(7)、他のプロセッサ基板に所有権が移転した

ことを示すが、キャッシュ状態タグメモリCTag₀のキャッシュ状態はIのままである。

【0189】次に、プロセッサ基板PB₁においては、キャッシュ状態タグメモリCTag₁がキャッシュ状態をIからMに移移し(8)、共有バスSBを介して転送されたデータがキャッシュメモリC₁にキャッシュされ(9)、キャッシュメモリC₁はキャッシュ状態をIからMに移移する(9)。このキャッシュC₁に対してCPU(CPU₁)が書き込みを行う。

10 【0190】図38は、プロセッサ基板PB₂のCPU(CPU₂)が、プロセッサ基板PB₁上のキャッシュメモリC₁で更新されて、その所有権がプロセッサ基板PB₀からプロセッサ基板PB₁に移転しているアドレス0x00001000に対して読み出しを行った(1)ときの動作を示している。プロセッサ基板PB₂のキャッシュメモリC₂は読み出しミスを起こし(2)、内部バスB₂に共有読み出しCRを発行する(3)。このアドレスはプロセッサ基板PB₂上の主メモリM₂に割り当てられたものではないので、プロセッサ基板PB₂の共有管理部S₂は共有バスSBに共有読み出しRSを発行する(4)。

30 【0191】プロセッサ基板PB₀からPB₁とPB₂は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板PB₀とPB₂はキャッシュ状態タグCTag₀、CTag₂が無効状態Iなので、動作を終了する(5)。特に、プロセッサ基板PB₀はこのアドレスが割り当てられた主メモリM₀を持つが、主記憶タグメモリMTag₀が状態Aで所有権の不在を示していることから内部バスB₀には一斉コマンドを発行しない。

【0192】一方、プロセッサ基板PB₁はキャッシュ状態タグCTag₁が状態Mなので、その内部バスB₁に共有読み出しCRを発行して(5)、キャッシュメモリC₁からこのアドレスの最新の値を読み出す(6)。この時点で、キャッシュメモリC₁はキャッシュ状態をMからOに移移し(6)、キャッシュ状態タグメモリCTag₁もキャッシュ状態をMからOに移移する(7)。読み出された最新の値は共有バスSBを介してプロセッサ基板PB₂に転送される(7)。

40 【0193】次に、プロセッサ基板PB₂においては、キャッシュ状態タグメモリCTag₂がキャッシュ状態をIからSに移移し(8)、共有バスSBを介して転送されたデータがキャッシュメモリC₂にキャッシュされて、キャッシュメモリC₂はキャッシュ状態をIからSに移移する(9)。このキャッシュC₂からCPU(CPU₂)に最新の値が読み出される。

50 【0194】図39はプロセッサ基板PB₁のCPU(CPU₁)が、プロセッサ基板PB₁上のキャッシュメモリで更新されて、その所有権がプロセッサ基板PB₀からプロセッサ基板PB₁に移転しているアドレス0x

0000 1000 に対してキャッシュの衝突又はキャッシュフ
ラッシュ命令に基づいて書き戻し(リプレース)を行っ
た(1)ときの動作を示している。プロセッサ基板PB₁
のキャッシュメモリC₁は最新の値を読み出し

(2)、キャッシュメモリC₂はキャッシュ状態をMからIに移移し、内部バスB₁に書き戻しWBを発行する

(3)。このアドレスはプロセッサ基板PB₁上の主メモリM₁に割り当てられたものではないので、プロセッサ基板PB₁の共有管理部S₁は共有バスSBに書き戻しWBを発行して(4)最新の値を転送すると共に、キャッシュ状態タグメモリCTag₁のキャッシュ状態をMからIに移移する。

[0195] プロセッサ基板PB₀とPB₂からPB₁は共有バスSB上の書き戻しWBを見て動作を開始するが、プロセッサ基板PB₁とPB₂はこのキャッシュラインを書き戻すエントリを主メモリM₂、M₃に持たないので動作を終了する(5)。プロセッサ基板PB₀はこのアドレスが割り当てられた主メモリM₀を持つので、その内部バスB₀に書き戻しWBを発行し(5)、主記憶タグメモリMTag₀の共有状態をAからGに移移して、所有権を取り戻したことを示す。共有バスSBを介して転送された最新の値が主メモリM₀に書き戻される(6)。

[0196] 図40は、プロセッサ基板PB₀のCPU(CPU₀)がその主メモリM₀に割り当てられたアドレス0x0000 1000に読み出しを行った(1)ときの動作を示している。プロセッサ基板PB₀のキャッシュメモリC₀は読み出しミスを起こし(2)、内部バスB₀に共有読み出しCRを発行する(3)。主記憶タグメモリMTag₀の共有状態がPで(4)、完全に排他なprivate状態であることを示している(5)ので共有バスSBにアクセスすることなく、主メモリM₀に直接アクセスを行い、このアドレスのデータを読み出し(4)、これをキャッシュメモリC₀にキャッシュする(5)。このキャッシュC₀からCPU₀がこのキャッシュラインを読み出す。

[0197] これに伴い、キャッシュメモリC₀をキャッシュ状態タグメモリCTag₀のキャッシュ状態がIからEに移移する(5)。

[0198] その後、このアドレスに対するCPU(CPU₀)からのアクセスはキャッシュメモリC₀でキャッシュヒットするので、内部バスB₀にコマンドを出すことなく行えるようになる。

[0199] 次に、本発明の第5実施例の分散共有メモリ型マルチプロセッサシステムについて説明する。

[0200] この第5実施例は上記第1実施例の改良である上記第4実施例を更に改良して、主記憶タグメモリ27で用いる共有状態の意味を変えて他のプロセッサ基板上のキャッシュメモリで書き換えられているか否か、即ち他のプロセッサ基板上に所有権が存在するか否か、

のみを示す2状態にしたものである。

[0201] このように共有状態を2状態に限定することにより、システムのコスト低減を図ることが可能となる。例えば主メモリ3が64メガ・バイトの場合主記憶タグを64バイトのキャッシュライン単位に持つとすると、共有状態が2状態ならば1ビットで表すことが可能であるから、全体として1メガ・ビットの主記憶タグメモリで充分であるが、共有状態が第4実施例の様に3状態の場合は2ビットで表さねばならず、全体として2メガ・ビットの主記憶タグメモリが必要となる。この容量は現在のSRAM1チップ分の容量に匹敵し、実現は可能であるが、コストの増大を免れ得ない。

[0202] この第5実施例では主記憶タグメモリ27は以下の2状態H、Aのいずれかを取り得る。

[0203] 「H」：他のプロセッサ基板のキャッシュにコピーが存在する可能性があるが、所有権はこのプロセッサ基板にある。

[0204] 「A」：他のプロセッサ基板のキャッシュに所有権がある。

[0205] この第5実施例においては、共有バス6からの介入を待つことなく所有権の所在が主記憶タグメモリ27の共有状態から前もって知ることが可能なため、指定されたアドレスを割り当てられた主メモリ3を持つプロセッサ基板での動作を主記憶タグメモリ27の共有状態に応じて以下の様にすることで、共有バス6上のトラフィックを減らすことが可能である。

[0206] 「H」：読み出しは内部バス5のみで行い、書き込みは共有バス6に無効化要求INVを発行して行う。

[0207] 「A」：読み出しは主メモリ3へのアクセスを中断して共有バス6からの介入を待って行き、書き込みは共有バス6に無効化要求INVを発行して行う。

[0208] 但し、状態Hにおいては読み出しを内部バス5のみで行うため、他のプロセッサ基板のキャッシュにコピーが存在するか否か正確な情報が得られない。このため、キャッシュのリフィル時に共有指示線を駆動して共有状態にあるものとしてキャッシュメモリに供給する必要がある。

[0209] 又、共有バス6上に自分の主メモリ3に割り当てられたアドレスに対するアクセス要求があっても、主記憶タグメモリ27の共有状態がAであれば他のプロセッサ基板が介入を行うもので、主メモリ3へのアクセスは不要であり内部バス5を一斉使わずに済むので、この内部バス5のトラフィックも削減出来る。

[0210] 従って、共有管理部4のキャッシュ状態タグメモリ24の状態遷移は自分のプロセッサ基板上の主メモリへのアクセスの際には上記図7にまとめたものに代わって図41にまとめたものとなり、他のプロセッサ基板上の主メモリへのアクセスの際には上記図8にまとめたものに代わって図42にまとめたものとなる。

【0211】図43はプロセッサ基板PB₀のCPU(CPU₀)が、その主メモリM₀に割り当てられたアドレス0x0000 1000に読み出しを行った(1)ときの動作を示している。プロセッサ基板PB₀のキャッシュメモリC₀は読み出しミスを起こし(2)、内部バスB₀に共有読み出しCRを発行し(3)、主メモリM₀にアクセスする(4)。主記憶タグメモリMTag₀の共有状態がHで(4)、他のプロセッサ基板に所有権が不在であることを示している(5)ので、共有バスSBにアクセスすることなく、主メモリM₀に直接アクセスを行い、このアドレスのデータを読み出し(4)、これをキャッシュメモリC₀にキャッシュする(5)。このキャッシュC₀からCPU₀がこのキャッシュラインを読み出す。

【0212】これに伴い、キャッシュメモリC₀とキャッシュ状態タグメモリCTag₀のキャッシュ状態がIからSに移移する。

【0213】その後、このアドレスに対するCPU(CPU₀)からのアクセスはキャッシュメモリC₀でキャッシュヒットするので、内部バスB₀にコマンドを出すことなく行えるようになる。

【0214】図44はプロセッサ基板PB₁のCPU(CPU₁)が、プロセッサ基板PB₀上の主メモリM₀に記憶されているアドレス0x0000 1000に対して書き込みを行った(1)ときの動作を示している。プロセッサ基板PB₁のキャッシュメモリC₁は書き込みミスを起こし(2)、内部バスB₁に排他読み出しCRIを発行する(3)。このアドレスはプロセッサ基板PB₁上の主メモリM₁に割り当てられたものではないので、プロセッサ基板PB₁の共有管理部S₁は共有バスSBに排他読み出しRMを発行する(4)。

【0215】プロセッサ基板PB₀とPB₁からPB₁は共有バスSB上の排他読み出しRMを見て動作を開始するが、プロセッサ基板PB₀とPB₁はこのキャッシュラインのコピーをキャッシュメモリC₀、C₁にも主メモリM₀、M₁にも持たないので、動作を終了する(5)。プロセッサ基板PB₀はこのアドレスが割り当てられた主メモリM₀を持つので、その内部バスB₀に排他読み出しCRIを発行して(5)、主メモリM₀からこのキャッシュラインを読み出す(6)。

【0216】読み出されたデータは共有バスSBを介してプロセッサ基板PB₁に転送される(7)。この時点で主記憶タグメモリMTag₀の共有状態はHからAに移移して、他のプロセッサ基板に所有権が移移したことを示すが、キャッシュ状態タグメモリCTag₀のキャッシュ状態はIのままである。

【0217】次に、プロセッサ基板PB₁においては、キャッシュ状態タグメモリCTag₁のキャッシュ状態がIからMに移移し(8)、共有バスSBを介して転送されたデータがキャッシュメモリC₁にキャッシュされ

(9)、キャッシュメモリC₁のキャッシュ状態がIからMに移移する(9)。このキャッシュC₁に対してCPU(CPU₁)が書き込みを行う。

【0218】図45はプロセッサ基板PB₀のCPU(CPU₀)が、プロセッサ基板PB₁上のキャッシュメモリC₁で更新されて、その所有権がプロセッサ基板PB₀からプロセッサ基板PB₁に移移しているアドレス0x0000 1000に対して読み出しを行った(1)ときの動作を示している。プロセッサ基板PB₀のキャッシュメモリC₀は読み出しミスを起こし(2)、内部バスB₀に共有読み出しCRを発行し(3)、主メモリM₀にアクセスする(4)。

【0219】しかし、主記憶タグメモリMTag₀の共有状態がAで(4)、このプロセッサ基板に所有権が不在であることを示している(5)ので、このプロセッサ基板PB₀の共有管理部S₀が共有バスSBの介入指示線を調べることなく主メモリM₀へのアクセスに介入して(5)、このアクセスを中断(6)する一方、共有バスSBに共有読み出しRSを発行する(4)。

【0220】プロセッサ基板PB₁からPB₀は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板PB₀とPB₁はこのキャッシュラインのコピーをキャッシュメモリC₀、C₁にも主メモリM₀、M₁にも持たないので動作を終了する(5)。プロセッサ基板PB₁は、キャッシュ状態タグCTag₁が状態Mなので、その内部バスB₁に共有読み出しCRを発行して(5)、キャッシュメモリC₁からこのアドレスの最新の値を読み出す(6)。この時点で、キャッシュメモリC₁のキャッシュ状態がMからOに移移し

(6)、キャッシュ状態タグメモリCTag₁のキャッシュ状態もMからOに移移する(7)。読み出された最新の値は共有バスSBを介してプロセッサ基板PB₀に転送される(7)。

【0221】次に、プロセッサ基板PB₀においては、キャッシュ状態タグメモリCTag₀のキャッシュ状態がIからSに移移する(8)が、主記憶タグメモリMTag₀の共有状態はAのままで、このプロセッサ基板に所有権が不在であることを示す。

【0222】共有バスSBを介して転送されたデータがキャッシュメモリC₀にキャッシュされて(9)、キャッシュメモリC₀のキャッシュ状態がIからSに移移する(9)。このキャッシュC₀からCPU(CPU₀)に最新の値が読み出される。

【0223】図46はプロセッサ基板PB₁のCPU(CPU₁)が、プロセッサ基板PB₁上のキャッシュメモリC₁で更新されて、その所有権がプロセッサ基板PB₀からプロセッサ基板PB₁に移移しているアドレス0x0000 1000に対して読み出しを行った(1)ときの動作を示している。プロセッサ基板PB₁のキャッシュメモリC₁は読み出しミスを起こし(2)、内部バスB₁

に共有読み出しCRを発行する(3)。このアドレスはプロセッサ基板PB₀上の主メモリM₀に割り当てられたものではないので、プロセッサ基板PB₀の共有管理部S₀は共有バスに共有読み出しRSを発行する(4)。

【0224】プロセッサ基板PB₀からPB₁とPB₂は共有バスSB上の共有読み出しRSを見て動作を開始するが、プロセッサ基板PB₀はこのキャッシュラインのコピーをキャッシュメモリC₀にも主メモリM₀にも持たないので動作を終了し(5)、プロセッサ基板PB₀は主記憶タグメモリMTag₀が状態Aであるので動作を終了する(5)。ここで、プロセッサ基板PB₀はこのアドレスが割り当てられた主メモリM₀を持つが、主記憶タグメモリMTag₀が状態Aで所有権の不在を示していることから内部バスB₀には一斉コマンドを発行しない。

【0225】一方、プロセッサ基板PB₁はキャッシュ状態タグCTag₁が状態Mなので、その内部バスB₁に共有読み出しCRを発行して(5)、キャッシュメモリC₁からこのアドレスの最新の値を読み出す(6)。この時点で、キャッシュメモリC₁のキャッシュ状態がMからOに遷移し(6)、キャッシュ状態タグメモリCTag₁のキャッシュ状態もMからOに遷移する(7)。読み出された最新の値は共有バスSBを介してプロセッサ基板PB₂に転送される(7)。

【0226】次に、プロセッサ基板PB₂においては、キャッシュ状態タグメモリCTag₂のキャッシュ状態がIからSに遷移し(8)、共有バスSBを介して転送されたデータがキャッシュメモリC₂にキャッシュされて(9)、キャッシュメモリC₂のキャッシュ状態がIからSに遷移する(9)。このキャッシュC₂からCPU(CPU₂)に最新の値が読み出される。

【0227】

【発明の効果】以上のように本発明によれば、同一基板上のメモリに対する非共有変数のアクセスを、共有バスを用いることなくかつ一貫性を保証しながら実現することができる。この同一基板上のメモリに対する非共有変数のアクセスは、通常メモリアccessの大部分を占めているため、CPUの実行動作性能の低下を防ぎ、また、共有バス上の不必要なトラフィックを削減できるのでより多くのプロセッサ基板をバス上に接続することが可能となる。結果として並列プログラムに共有変数アクセスに対する制約を加えることなく、高いシステム拡張性を持つマルチプロセッサシステムを実現できるという多大な効果を奏する。

【図面の簡単な説明】

【図1】従来の共有メモリ型マルチプロセッサシステムの概略ブロック図。

【図2】従来の分散共有メモリ型マルチプロセッサシステムの概略ブロック図。

【図3】本発明の第1実施例の分散共有メモリ型マルチプロセッサシステムの概略ブロック図。

【図4】本発明の第1実施例におけるキャッシュメモリ2の詳細ブロック図。

【図5】本発明の第1実施例における共有管理部4の詳細ブロック図。

【図6】本発明の第1実施例における状態タグ部12の状態とCPUからの命令及び内部バスからのコマンドに対する発行コマンド/状態遷移をまとめた表を示す図。

10 【図7】本発明の第1実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、自分の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図8】本発明の第1実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、他の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

20 【図9】本発明の第1実施例におけるCPUの読み出し命令に対するキャッシュ制御部及び共有管理部の動作を示すフローチャートの前半。

【図10】本発明の第1実施例におけるCPUの読み出し命令に対するキャッシュ制御部及び共有管理部の動作を示すフローチャートの後半。

【図11】本発明の第1実施例におけるCPUの書き込み命令に対するキャッシュ制御部及び共有管理部の動作を示すフローチャート。

30 【図12】本発明の第1実施例における共有読み出しに対する共有管理部及びキャッシュ制御部の動作を示すフローチャート。

【図13】本発明の第1実施例における排他読み出しに対する共有管理部及びキャッシュ制御部の動作を示すフローチャート。

【図14】本発明の第1実施例における無効化要求に対する共有管理部及びキャッシュ制御部の動作を示すフローチャート。

【図15】本発明の分散共有メモリ型マルチプロセッサシステムの構成例を示す図。

40 【図16】本発明の第1実施例の具体的動作例を図15の構成例において示す図。

【図17】本発明の第1実施例の具体的動作例を図15の構成例において示す図。

【図18】本発明の第1実施例の具体的動作例を図15の構成例において示す図。

【図19】本発明の第1実施例の具体的動作例を図15の構成例において示す図。

【図20】本発明の第1実施例の具体的動作例を図15の構成例において示す図。

50 【図21】本発明の第1実施例の具体的動作例を図15

の構成例において示す図。

【図22】本発明の第1実施例の具体的動作例を図15の構成例において示す図。

【図23】本発明の第2実施例における状態タグ部12の状態とCPUからの命令及び内部バスからのコマンドに対する発行コマンド/状態遷移をまとめた表を示す図。

【図24】本発明の第2実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、自分の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図25】本発明の第2実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、他の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図26】本発明の第2実施例におけるCPUの読み出し命令に対するキャッシュ制御部及び共有管理部の動作を示すフローチャートの前半。

【図27】本発明の第2実施例におけるCPUの読み出し命令に対するキャッシュ制御部及び共有管理部の動作を示すフローチャートの後半。

【図28】本発明の第2実施例における共有読み出しに対する共有管理部及びキャッシュ制御部の動作を示すフローチャート。

【図29】本発明の第2実施例の具体的動作例を図15の構成例において示す図。

【図30】本発明の第3実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、自分の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図31】本発明の第3実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、他の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図32】本発明の第3実施例の具体的動作例を図15の構成例において示す図。

【図33】本発明の第3実施例の具体的動作例を図15の構成例において示す図。

【図34】本発明の第3実施例の具体的動作例を図15の構成例において示す図。

【図35】本発明の第4実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、自分の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図36】本発明の第4実施例におけるキャッシュ状態

タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、他の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図37】本発明の第4実施例の具体的動作例を図15の構成例において示す図。

【図38】本発明の第4実施例の具体的動作例を図15の構成例において示す図。

【図39】本発明の第4実施例の具体的動作例を図15の構成例において示す図。

【図40】本発明の第4実施例の具体的動作例を図15の構成例において示す図。

【図41】本発明の第5実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、自分の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図42】本発明の第5実施例におけるキャッシュ状態タグメモリ24の状態と内部バス及び共有バスからのコマンドに対する発行コマンド/状態遷移を、他の基板上の主メモリに対するアクセスの場合についてまとめた表を示す図。

【図43】本発明の第5実施例の具体的動作例を図15の構成例において示す図。

【図44】本発明の第5実施例の具体的動作例を図15の構成例において示す図。

【図45】本発明の第5実施例の具体的動作例を図15の構成例において示す図。

【図46】本発明の第5実施例の具体的動作例を図15の構成例において示す図。

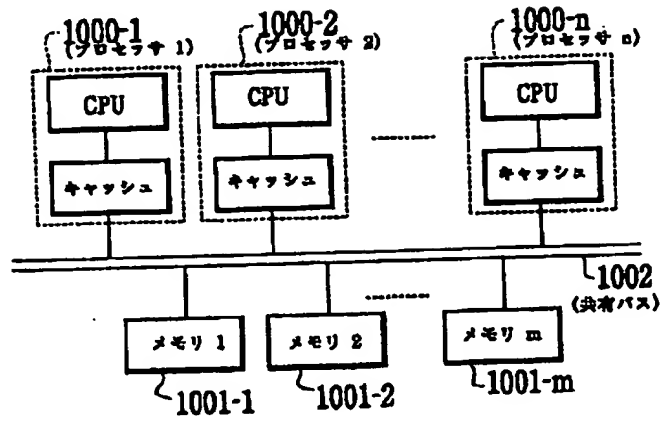
【符号の説明】

- 1 CPU
- 2 キャッシュメモリ
- 3 主記憶
- 4 共有管理部
- 5 内部バス
- 6 共有バス
- 11 アドレスタグ部
- 12 状態タグ部
- 13 データ記憶部
- 14 比較器
- 15 有効/無効判断部
- 16 hit/miss信号算出部
- 17 キャッシュ制御部
- 18 共有指示線
- 19 コマンド線
- 20 アドレス線
- 21 介入指示線
- 22 データ線
- 23 キャッシュアドレスタグメモリ

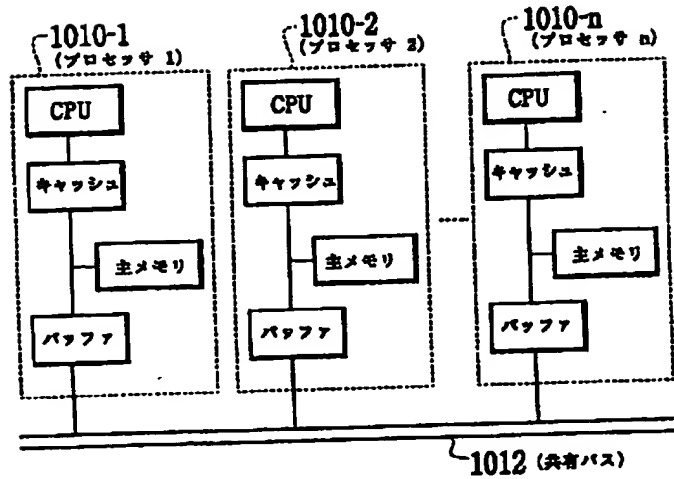
- 43
 24 キャッシュ状態タグメモリ
 25 キャッシュ状態タグメモリ制御部
 26 主記憶タグメモリ制御部

- 44
 27 主記憶タグメモリ
 28 外部アクセス制御部
 29 内部アクセス制御部

【図1】



【図2】



(24)

【図3】

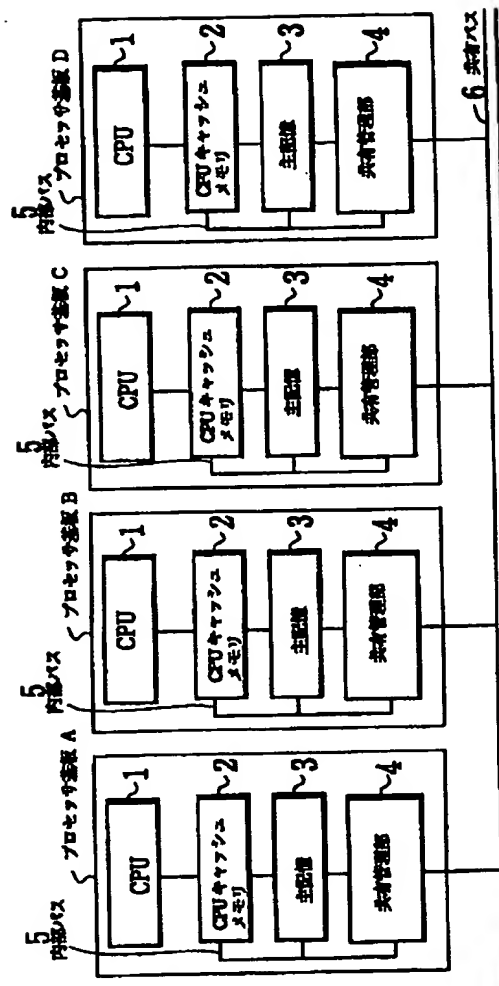
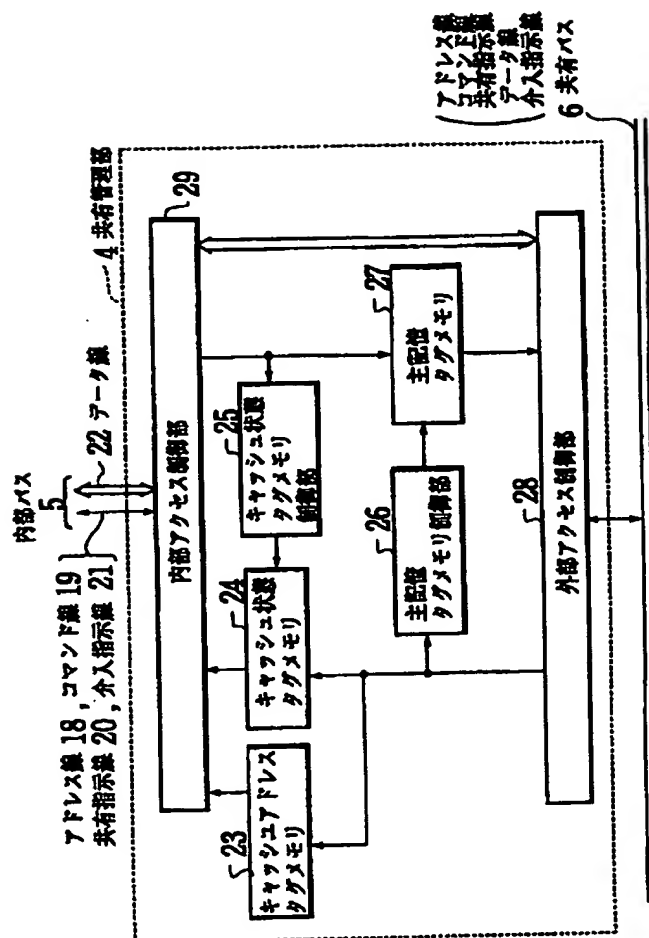


Figure 1 is a block diagram of a cache control system. The system includes a CPU (1) connected to a cache memory (2) via a data bus (8). The CPU also provides an address (7) to the system. The cache memory (2) is connected to a data bus (5) and a cache control unit (17). The cache control unit (17) receives a common control signal (18) and a cache hit/miss signal (16). It also outputs a cache hit/miss signal (16) to the cache memory (2). The cache control unit (17) is connected to a cache tag unit (11) and a status tag unit (12). The cache tag unit (11) receives an address (7) and outputs a hit/miss signal (16) to the cache control unit (17). The status tag unit (12) outputs a hit/miss signal (16) to the cache control unit (17). The cache control unit (17) also outputs a hit/miss signal (16) to the cache memory (2). The cache control unit (17) is connected to a cache memory (13) via a data bus (22). The cache memory (13) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (13). The cache control unit (17) is connected to a cache memory (14) via a data bus (22). The cache memory (14) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (14). The cache control unit (17) is connected to a cache memory (15) via a data bus (22). The cache memory (15) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (15). The cache control unit (17) is connected to a cache memory (16) via a data bus (22). The cache memory (16) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (16). The cache control unit (17) is connected to a cache memory (17) via a data bus (22). The cache memory (17) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (17). The cache control unit (17) is connected to a cache memory (18) via a data bus (22). The cache memory (18) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (18). The cache control unit (17) is connected to a cache memory (19) via a data bus (22). The cache memory (19) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (19). The cache control unit (17) is connected to a cache memory (20) via a data bus (22). The cache memory (20) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (20). The cache control unit (17) is connected to a cache memory (21) via a data bus (22). The cache memory (21) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (21). The cache control unit (17) is connected to a cache memory (22) via a data bus (22). The cache memory (22) outputs data (2) to the cache control unit (17). The cache control unit (17) also outputs data (2) to the cache memory (22).

【图5】



【図6】

キャッシュメモリ

(a)

CPUからの 状態データ	Read	Write
M	ヒット	ヒット
O	ヒット	CL/M
E	ヒット	ヒット/M
S	ヒット	CL/M
I	CR/E(S)	CR/M

内部バスの共有要求線が断線されているとき

(b)

内部バスからの 状態データ	CR	CR1	CI	WR
M	全入/O	全入/I	エラー	エラー
O	全入/O	全入/I	-/I	エラー
E	-/I	-/I	エラー	エラー
S	---	-/I	-/I	---
I	---	---	---	---

【図23】

キャッシュメモリ

(a)

CPUからの 状態データ	Read	Write
M	ヒット	ヒット
O	ヒット	CL/M
E	ヒット	ヒット/M
S	ヒット	CL/M
I	CR/S(S) (O)	CR/M

内部バスの共有要求線が断線されているとき

共有要求線(全入があるとき)

(b)

内部バスからの 状態データ	CR	CR1	CI	WR
M	全入/O	全入/I	エラー	エラー
O	全入/O	全入/I	-/I	エラー
E	-/I	-/I	エラー	エラー
S	---	-/I	-/I	---
I	---	---	---	---

【図7】

共有管理部(自分の基板上の主メモリに対するアクセスの場合)

(a)

内部バスからの 状態タグ24 コマンド		CR	CRI	CI	WR
I	M(P)	エラー	エラー	エラー	-/I・P
	O(G)	エラー	エラー	INV/M	-/I・G
	E(P)	——	-/M	エラー	-/I・P
	S(G)	RS/E(S)	RM/M	INV/M	エラー
	P	-/E	-/M	エラー	エラー
	G	RS/E(S)	RM/M	エラー	エラー

27
主記憶タグ共有バスの共有指示線が駆動されているとき
《内部バスの共有指示線を共有バスと同様に駆動する》

(b)

《共有バスの介入指示線
を駆動する》 《内部バス同様に共有バスの
介入指示線を駆動する》

共有バスからの 状態タグ24 コマンド		RS	RM	INV	WB
I	M(P)	CR/O	CRI/I・G	エラー	エラー
	O(G)	CR/O	CRI/I・G	CI/I・G	エラー
	E(P)	CR/O(S)	CRI/I・G	エラー	エラー
	S(G)	CR/S	CRI/I・G	CI/I・G	WR/S
	P	CR/I・G	CRI/I・G	エラー	エラー
	G	CR/I・G	CRI/I・G	——	WR/I・G

27
主記憶タグ介入有のとき(無のとき)
《共有バスの共有指示線を駆動する》

[図8]

共有管理部(他の基板上の主メモリに対するアクセスの場合)

(a)

内部バスからの 状態タグ24 コマンド	CR	CRI	CI	WR
M	エラー	エラー	エラー	WB/I
O	エラー	エラー	INV/M	WB/I
E	RS/E(S)	RM/M	-/M	エラー
S	RS/E(S)	RM/M	INV/M	エラー
I	RS/E(S)	RM/M	エラー	エラー

共有バスの共有指示線が駆動されているとき
 (共有バスの方が駆動されているか否かに関わらず内部バス
 の共有指示線を駆動する)

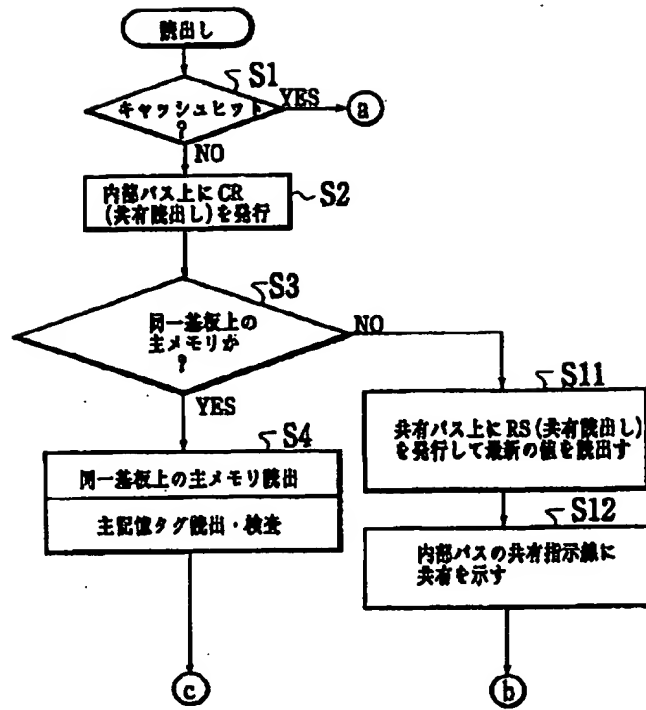
(b)

《共有バスの介入指示線を駆動する》

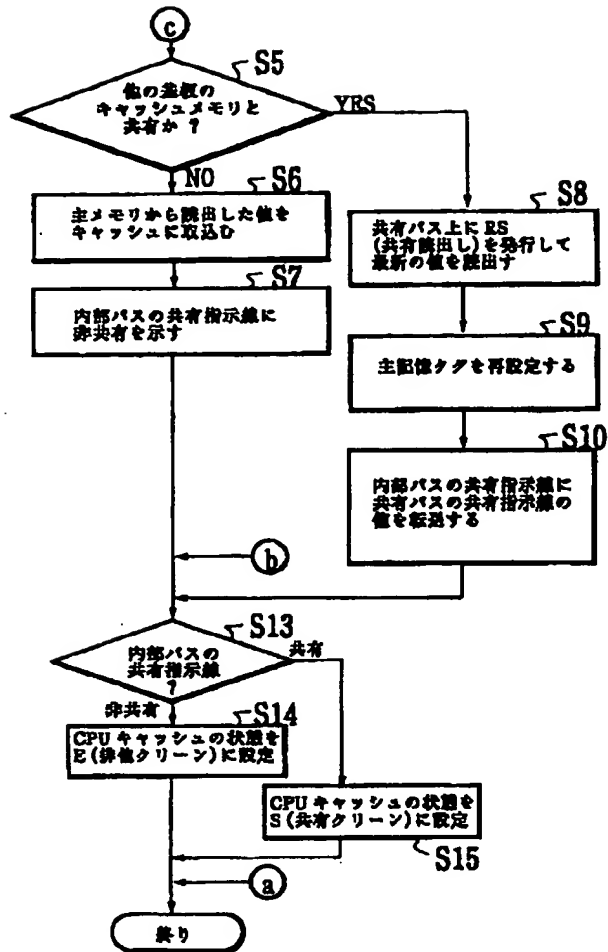
共有バスからの 状態タグ24 コマンド	RS	RM	INV	WB
M	CR/O	CRI/I	エラー	エラー
O	CR/O	CRI/I	CI/I	エラー
E	-/S	CI/I	エラー	エラー
S	—	CI/I	CI/I	—
I	—	—	—	—

《共有バスの共有指示線を駆動する》

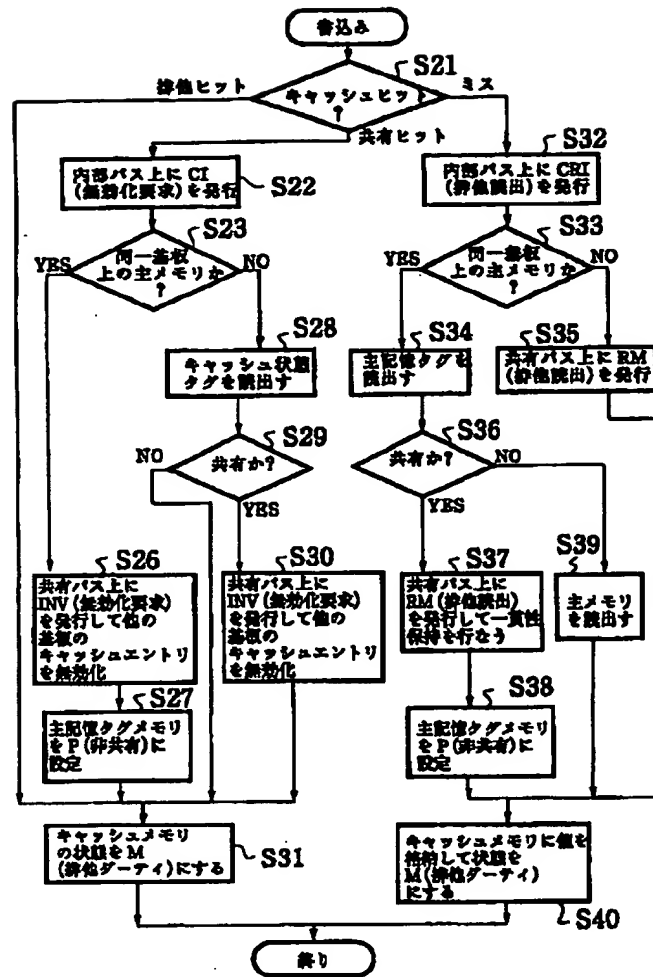
【図9】



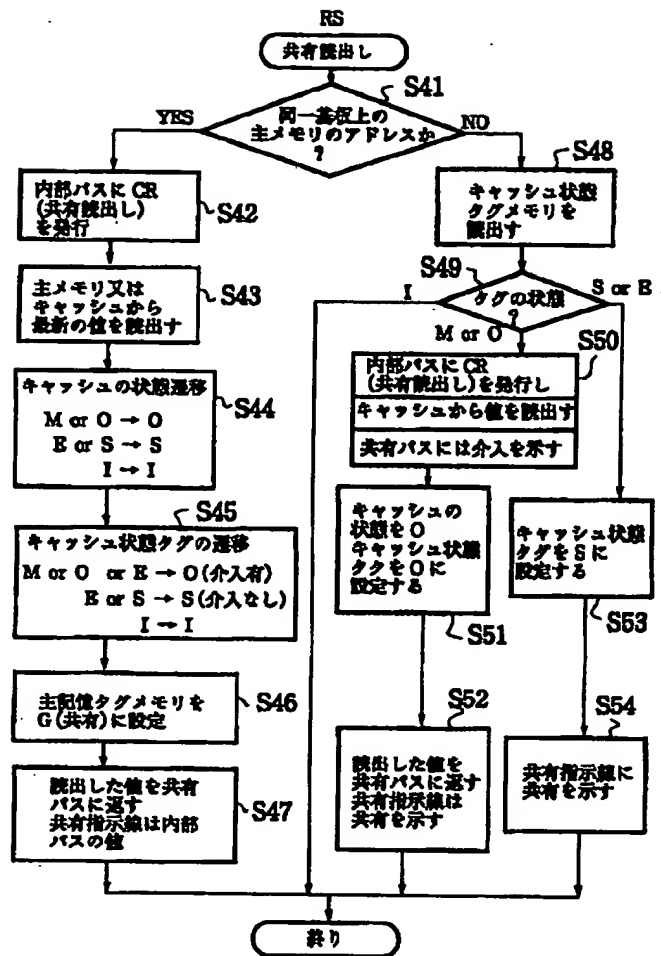
【図10】



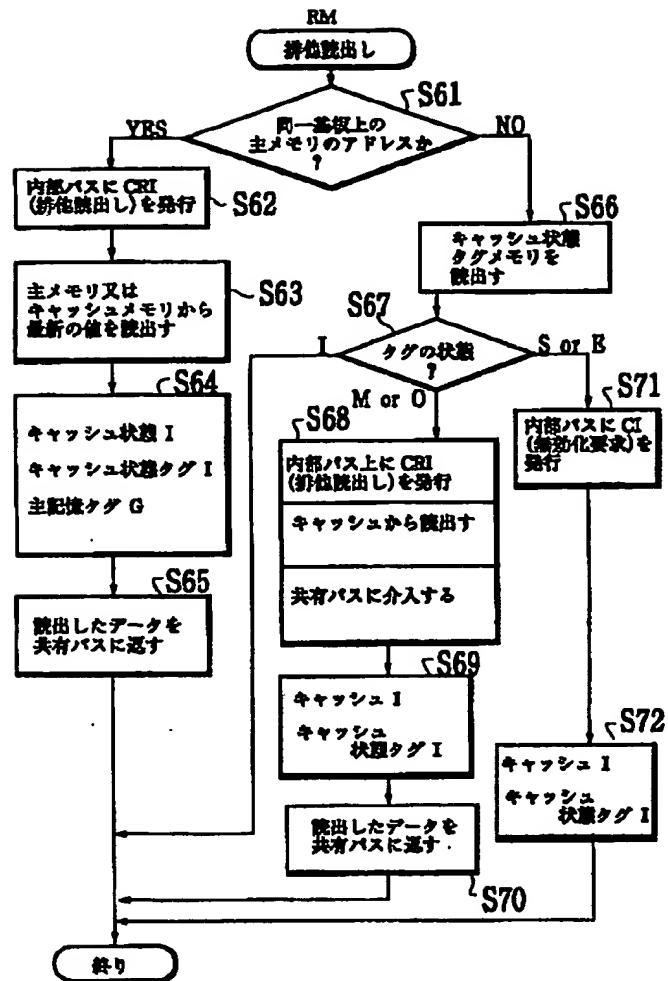
【図11】



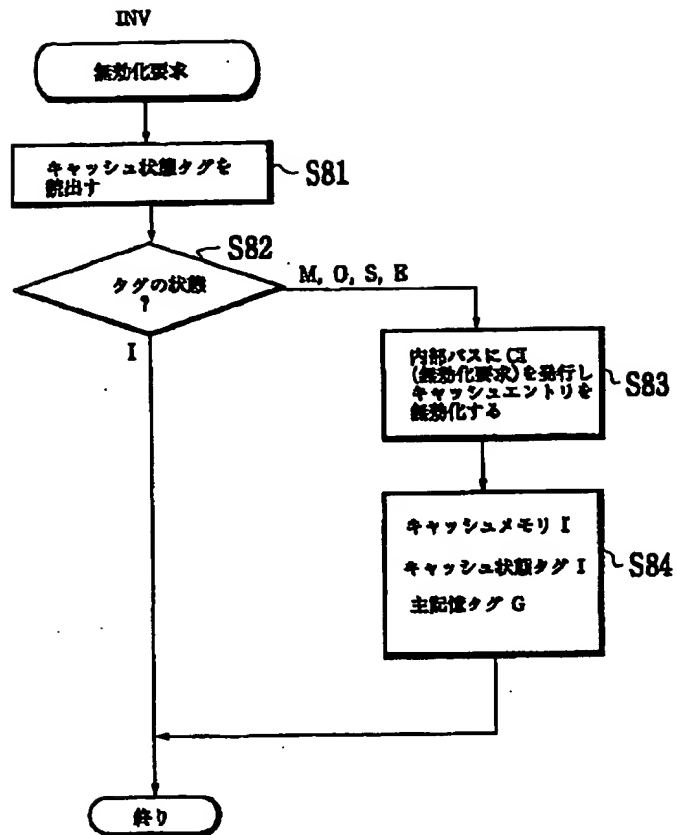
【図12】



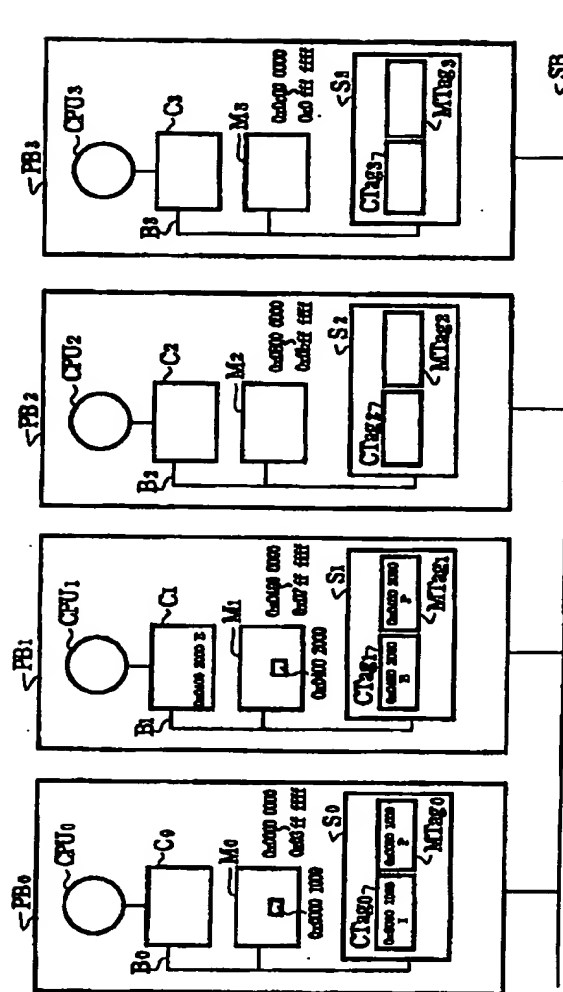
〔図13〕



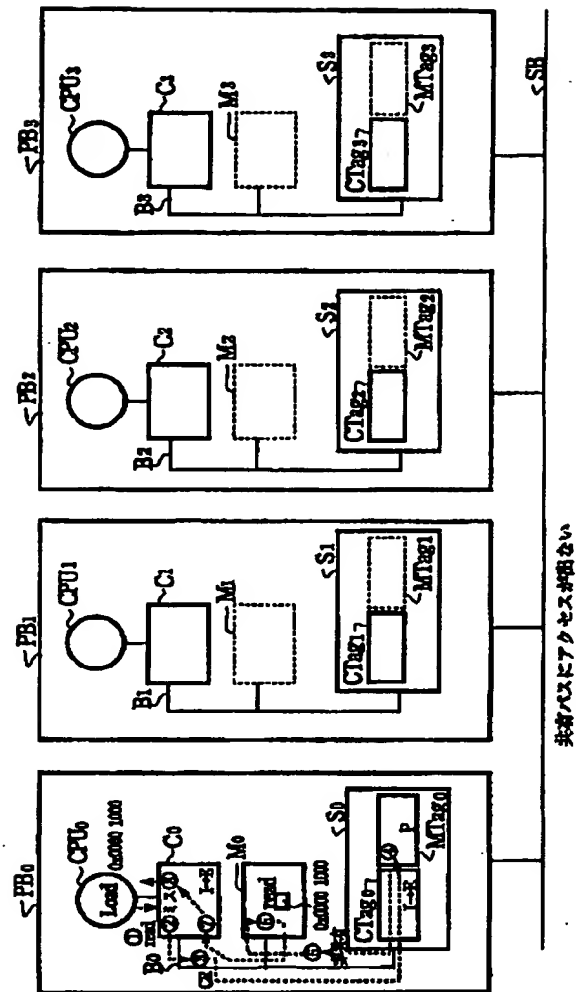
【図14】



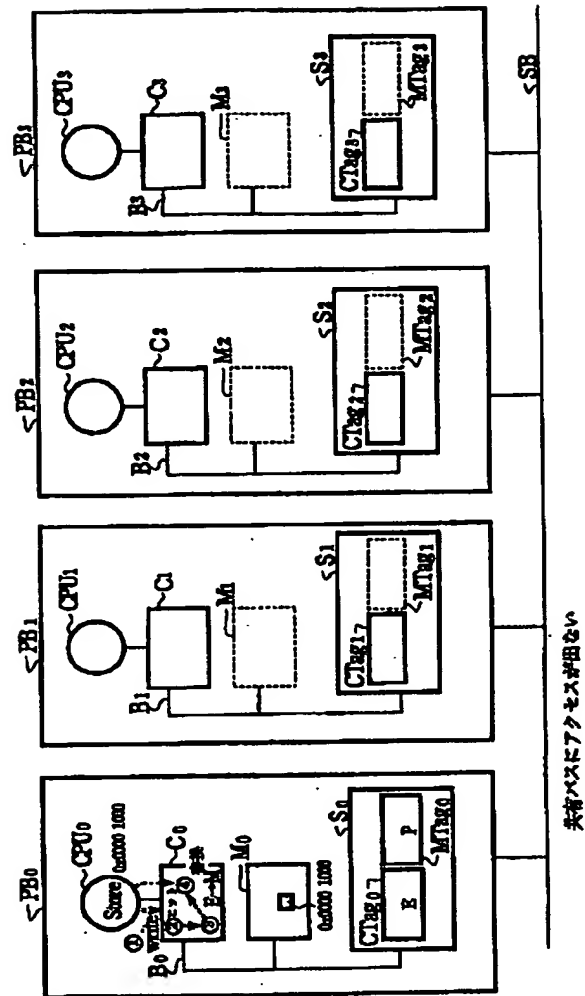
〔図15〕



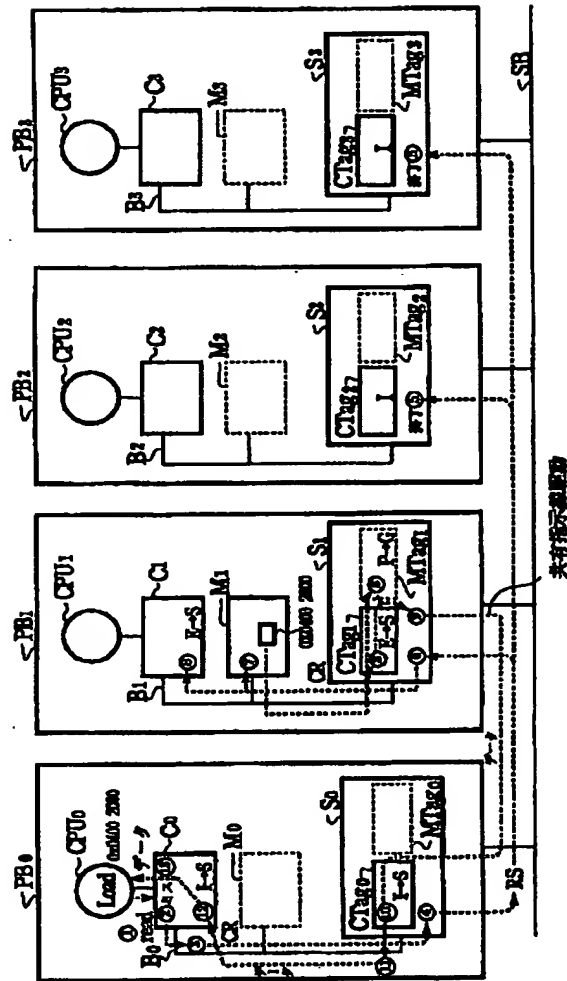
〔図16〕



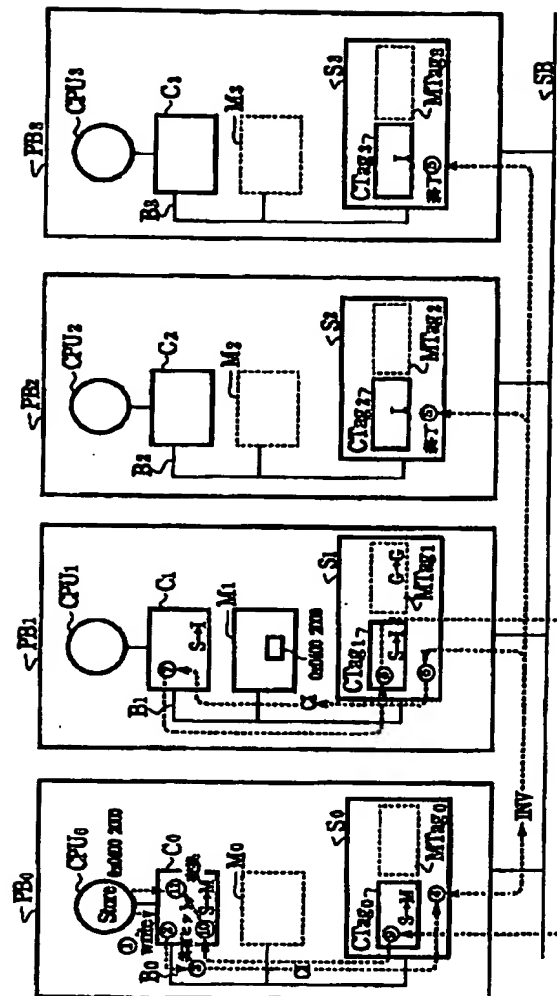
[図17]



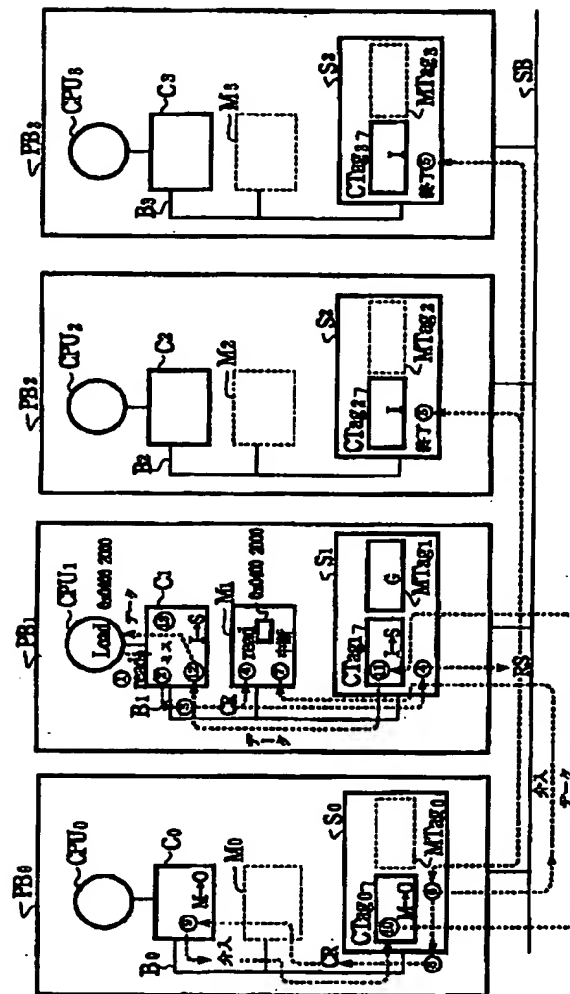
[図18]



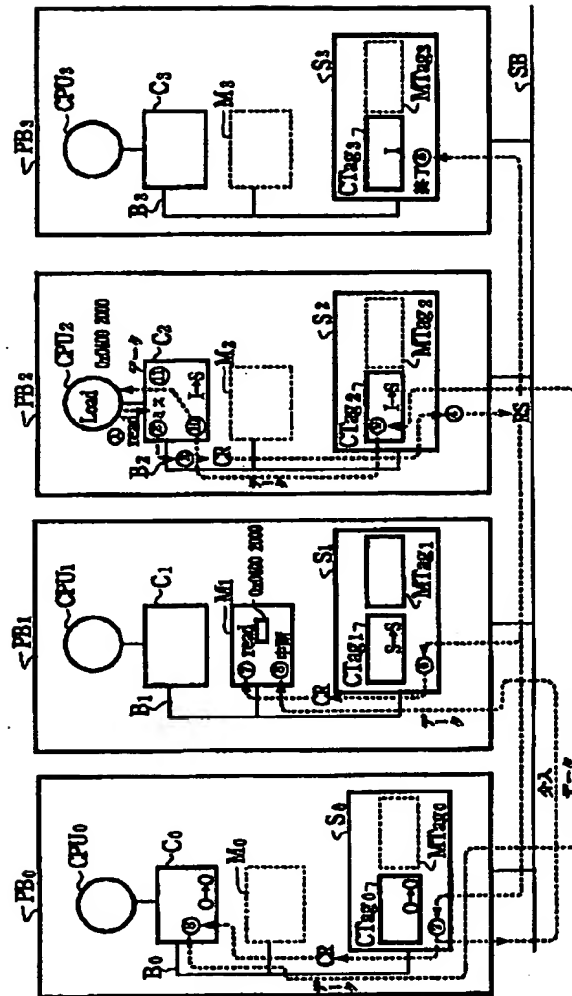
[図19]



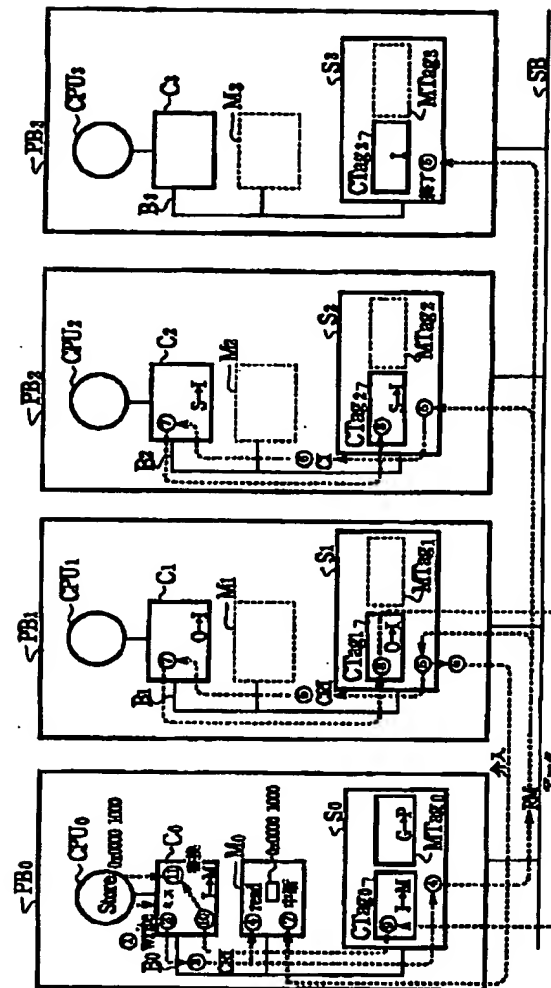
[図20]



[図21]



[図22]



【図24】

共有管理部(自分の基板上の主メモリに対するアクセスの場合)

(a)

内部バスからの 状態タグ24 コマンド		CR	CRI	CI	WR
M(P)		エラー	エラー	エラー	-/I・P
O(G)		エラー	エラー	INV/M	-/I・G
E(P)		——	-/M	エラー	-/I・P
S(G)		RS/E(S)(O)	RM/M	INV/M	エラー
I	P	-/E	-/M	エラー	エラー
	G	RS/E(S)(O)	RM/M	エラー	エラー

27
主記憶タグ

所有権移転(介入があったとき)

共有バスの共有指示線が駆動されて
いるとき(内部バスの共有指示
線を共有バスと同様に駆動する)

(b)

共有バスからの 状態タグ24 コマンド		RS	RM	INV	WB
M(P)		CR/O	CRI/I・G	エラー	エラー
O(G)		CR/O	CRI/I・G	CI/I・G	エラー
E(P)		CR/O(S)	CRI/I・G	エラー	エラー
S(G)		CR/S	CRI/I・G	CI/I・G	WR/S
I	P	CR/I・G	CRI/I・G	エラー	エラー
	G	CR/I・G	CRI/I・G	——	WR/I・G

27
主記憶タグ

介入有のとき(無のとき)

(共有バスの共有指示線を駆動する)

[図25]

共有管理部(他の基板上の主メモリに対するアクセスの場合)

(a)

内部バスからの 状態タグ24 コマンド	CR	CRI	CI	WR
M	エラー	エラー	エラー	WB/I
O	エラー	エラー	INV/M	WB/I
E	RS/E(S)	RM/M	-/M	エラー
S	RS/E(S)	RM/M	INV/M	エラー
I	RS/E(S)	RM/M	エラー	エラー

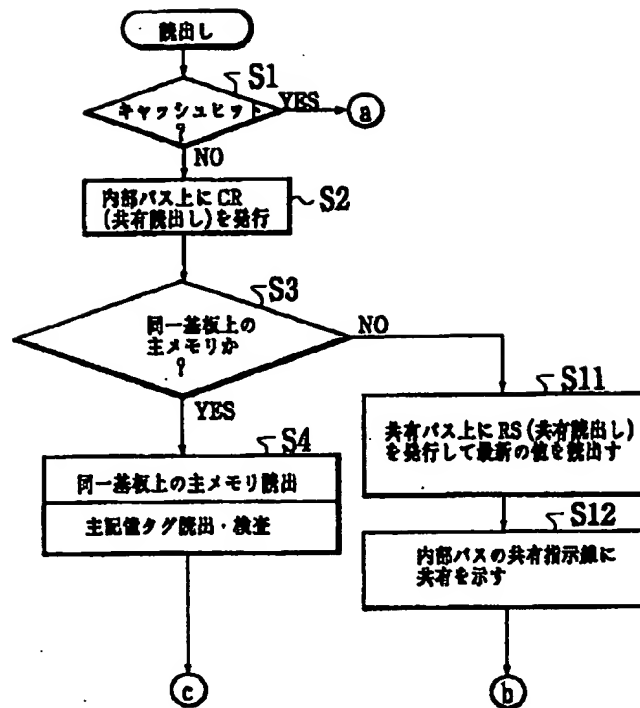
共有バスの共有指示線が駆動されているとき
 (共有バスの方が駆動されているか否かに関わらず
 内部バスの共有指示線を駆動する)

(b)

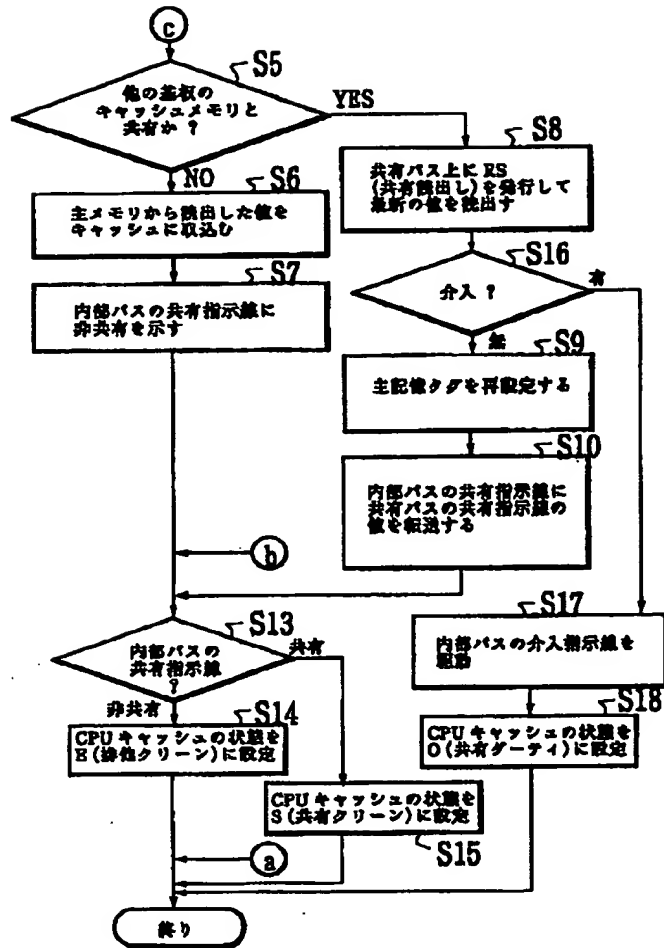
共有バスからの 状態タグ24 コマンド	RS	RM	INV	WB
M	CR/O(S)	CRI/I	エラー	エラー
O	CR/O(S)	CRI/I	CI/I	エラー
E	-/S	CI/I	エラー	エラー
S	—	CI/I	CI/I	—
I	—	—	—	—

所有権移転
 (共有バスの共有指示線を駆動する)

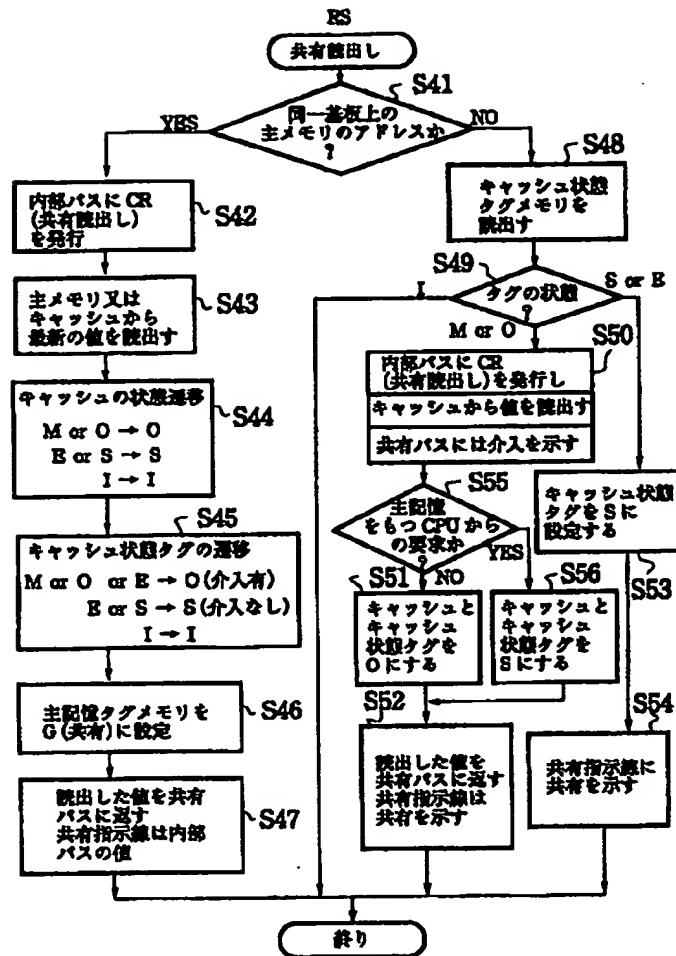
【図26】



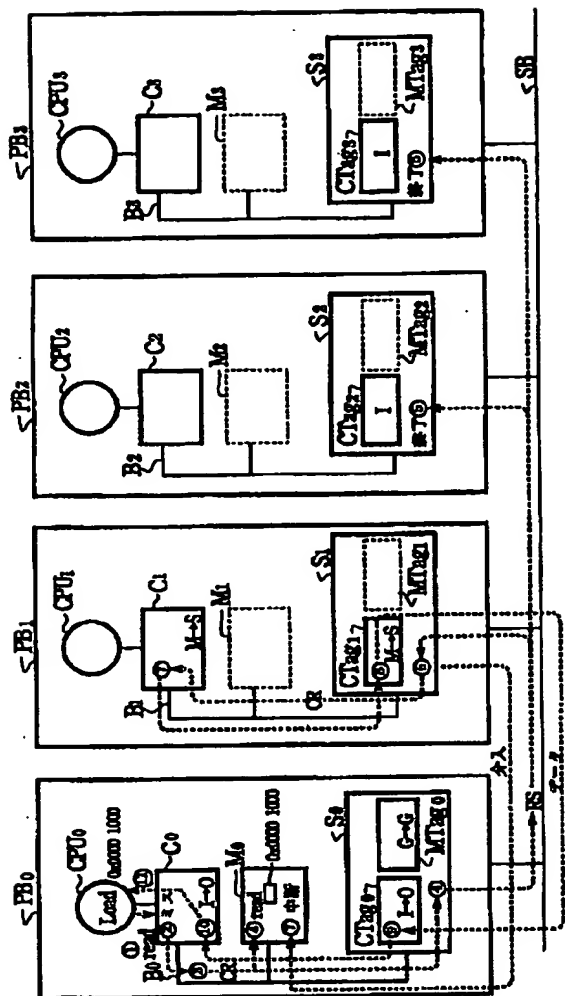
【図27】



[図28]



【図29】



[図30]

(a)

内部バスからの 状態タグ24 コマンド		CR	CRI	CI	WR
M(P)		エラー	エラー	エラー	-/I・P
E		-/E	-/M	エラー	-/I・P
S		RS/E(S)	RM/M	INV/M	エラー
I	P	-/E	-/M	エラー	エラー
	G	RS/E(S)	RM/M	エラー	エラー

共有バスの共有指示線が駆動されているとき

(b)

共有バスから 状態タグ24 コマンド		共有バスに介入する 共有バスの共有指示線を 駆動する		CPU キャッシュが介入すれば 共有バスに介入する	
		RS	RM	INV	WB
M(P)		CR/S	CRI/I・G	エラー	エラー
E		CR/S	CRI/I・G	エラー	エラー
S		CR/S	CRI/I・G	INV/I・G	エラー
I	P	CR/I・G	CRI/I・G	エラー	エラー
	G	CRI/I・G (WR/I・G)	CRI/I・G (WR/I・G)	-/I・G	WR/I・P

共有バスから介入があった時

(図31)

(a)

共有バスからの 受信データ	CR	CR1	CI	WR
M	エラ-	エラ-	エラ-	WR/1
E	RM/0	RM/M	-/M	エラ-
S	RM/0	RM/M	INV/M	エラ-
I	RM/0	RM/M	エラ-	エラ-

共有バスの共有要求値が置換されているとき

(b)

共有バスに代入する

共有バスからの 受信データ	RS	RM	INV	WB
M	CL/0	CL/1	エラ-	エラ-
E	-/0	CL/1	エラ-	エラ-
S	-/0	CL/1	CL/1	-/0
I	---	---	---	---

共有バスの共有要求値を置換する

(図35)

(a)

共有バスからの 受信データ	CR	CR1	CI	WR
M (P)	エラ-	エラ-	エラ-	-/1.0
O (G)	エラ-	エラ-	INV/M	-/1.0
E (P)	-/0	-/M	エラ-	-/1.0
S	G	-/0	-/M	INV/M
	A	RM/0	RM/M	INV/M
I	P	-/0	-/M	エラ-
	G	-/0	INV/M	エラ-
	A	RM/0	RM/M	エラ-

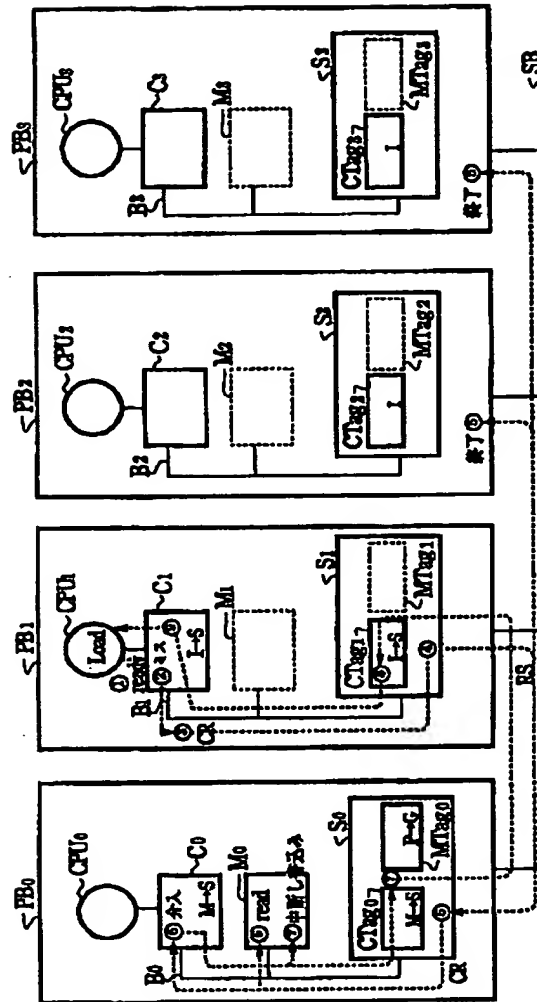
(b)

共有バスに代入する CPU キャッシュに代入
すれば共有バスに代入する

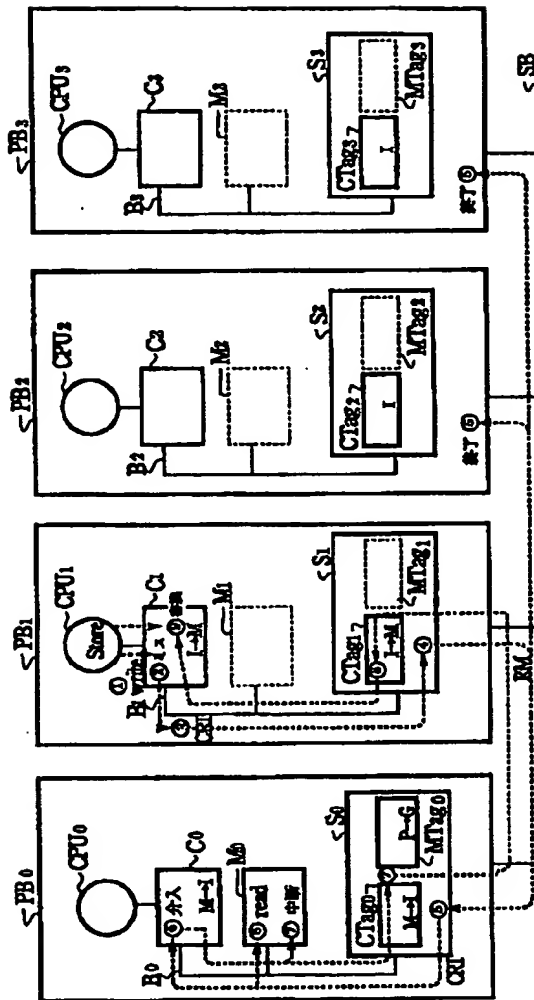
共有バスからの 受信データ	RS	RM	INV	WB
M (P)	CL/0	CL/1	エラ-	エラ-
O (G)	CL/0	CL/1	CL/1	エラ-
E (P)	CL/0	CL/1	エラ-	エラ-
S	G	CL/0	CL/1	WR/0
	A	-/0	CL/1	WR/0
I	P	CL/0	CL/1	エラ-
	G	CL/0	CL/1	-/0
	A	-/0	-/1	WR/0

共有バスの共有要求値を置換する

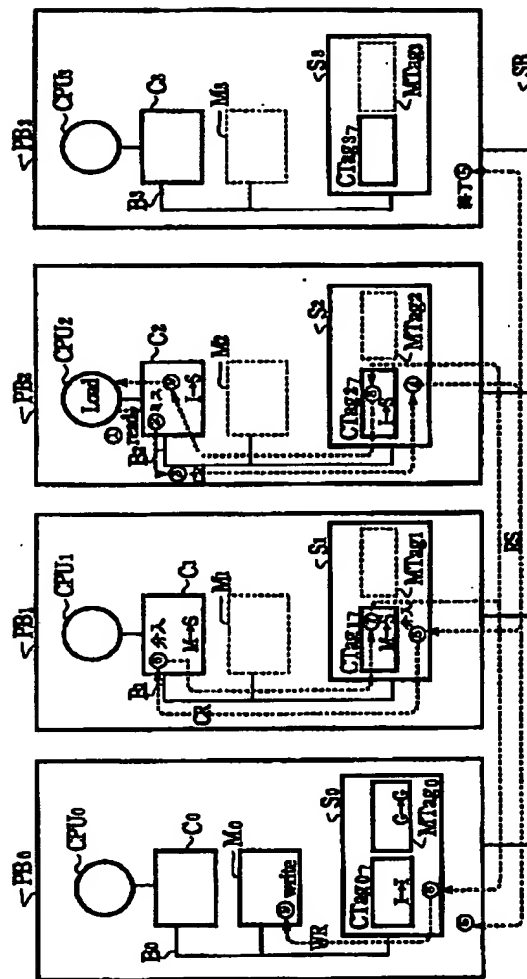
[図32]



[図33]



[図34]



[図36]

(a)

共有バスからの 受信データ	CR	CR1	CI	WR
M	エ9-	エ9-	エ9-	WR/1
O	エ9-	エ9-	INV/M	WR/1
E	EA/EO	EA/M	-/M	エ9-
S	EA/EO	EA/M	INV/M	エ9-
I	EA/EO	EA/M	エ9-	エ9-

共有バスの共有データが読取されているとき

(b)

共有バスに代入する

CPU キャッシュが代入すれば
共有バスに代入する

共有バスからの 受信データ	RS	RM	INV	WB
M	EA/EO	CR/1	エ9-	エ9-
O	EA/EO	CR/1	CR/1	エ9-
E	-/1	CR/1	エ9-	エ9-
S	-/1	CR/1	CR/1	-/1
I	---	---	---	---

共有バスの共有データが読取する

[図41]

(a)

共有バスからの 受信データ	CR	CR1	CI	WR
M(H)	エ9-	エ9-	エ9-	-/1・H
O(H)	エ9-	エ9-	INV/M	-/1・H
E(H)	-/1	-/1	エ9-	-/1・H
S	H	-/1・H	INV/M	INV/M
	A	EA/1・A	EA/M	INV/M
I	H	-/1・H	INV/M	エ9-
	A	EA/1・A	EA/M	エ9-

(b)

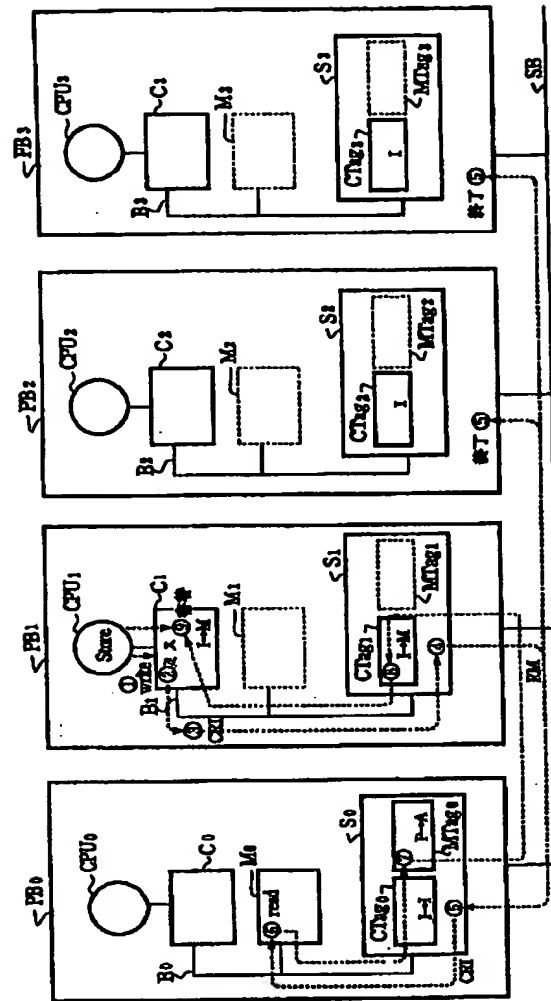
共有バスに代入する

CPU キャッシュが代入すれば
共有バスに代入する

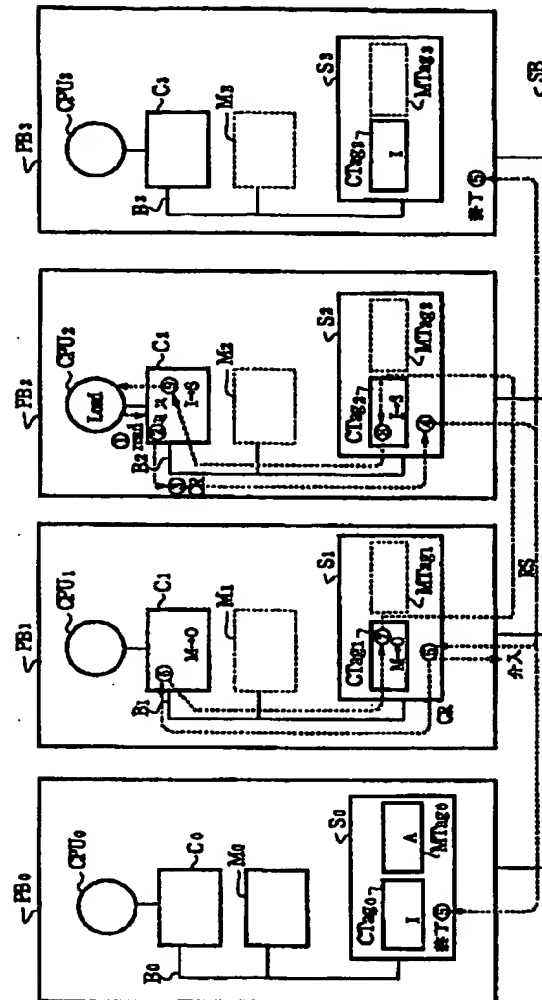
共有バスからの 受信データ	RS	RM	INV	WB
M(H)	CR/0	CR/1・A	エ9-	エ9-
O(H)	CR/0	CR/1・A	CR/1・A	エ9-
E(H)	CR/0・H	CR/1・A	エ9-	エ9-
S	H	CR/1・H	CR/1・A	CR/1・A
	A	-/1・A	CR/1・A	CR/1・A
I	H	CR/1・H	CR/1・A	-/1・A
	A	-/1・A	-/1・A	WR/1・H

共有バスの共有データが読取する

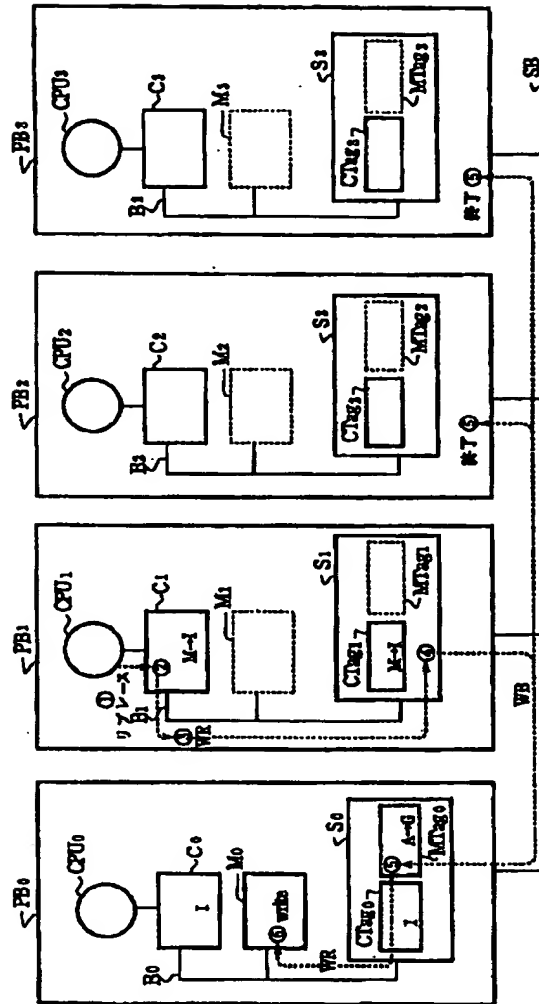
〔図37〕



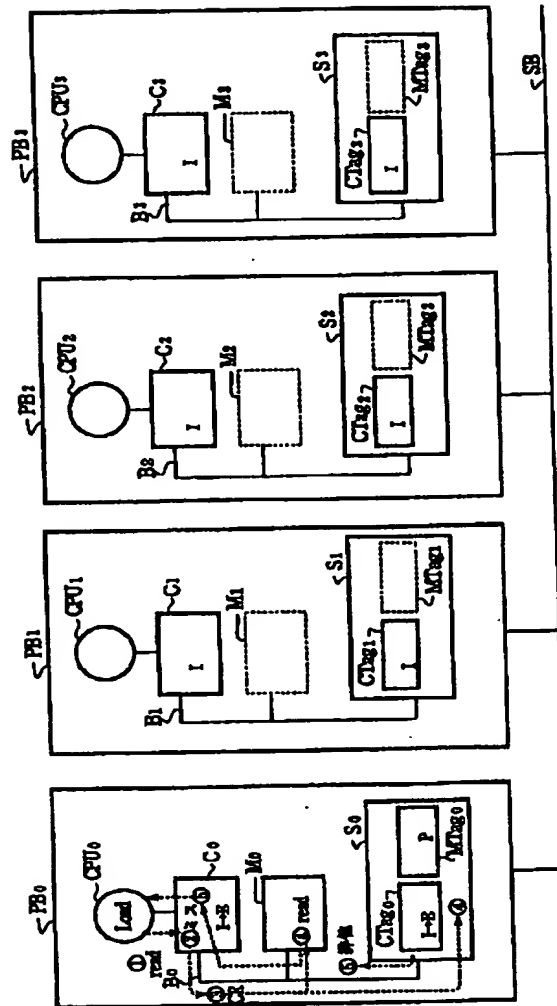
(図38)



[図39]



[図40]



〔図42〕

(a)

共有バスからの 装置アドレス コマンド	CR	CRJ	CI	WR
M	エラー	エラー	エラー	WR/1
O	エラー	エラー	INT/M	WR/1
R	RM/M	RM/M	-/M	エラー
S	RM/M	RM/M	INT/M	エラー
I	RM/M	RM/M	エラー	エラー

共有バスの共有装置が故障している時

(b)

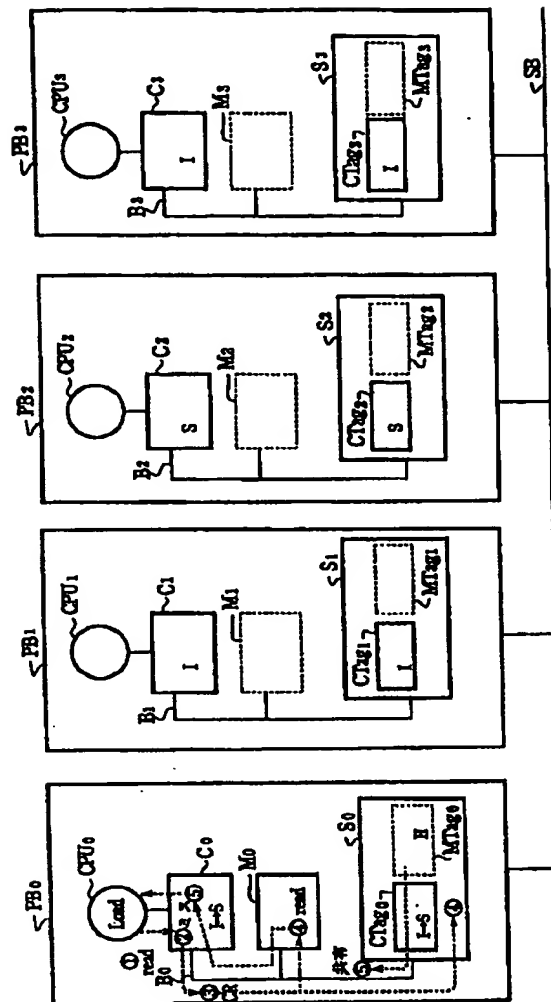
共有バスに介入する

CPUがマスタ介入すれば
共有バスに介入する

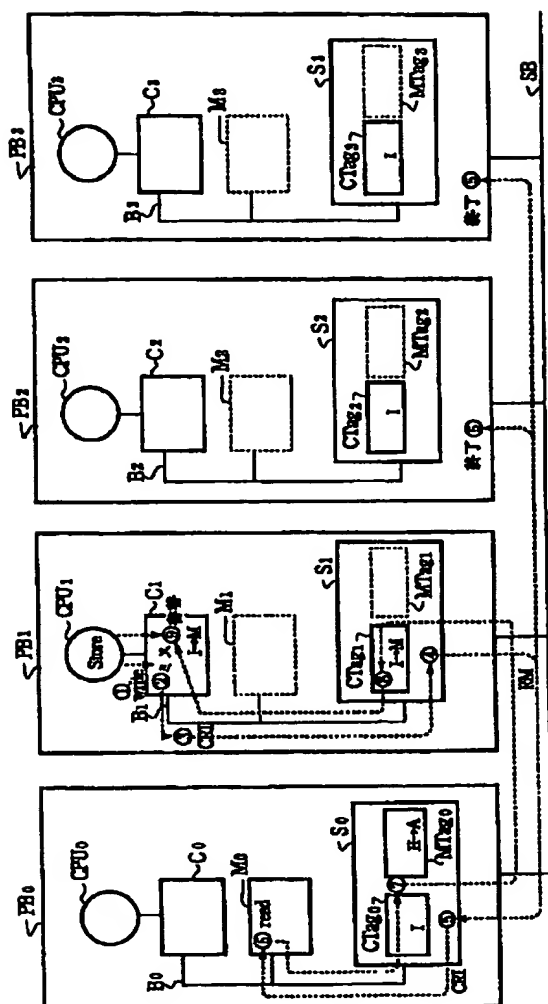
共有バスからの 装置アドレス コマンド	RS	RM	INV	WB
M	CR/0	CR/1	エラー	エラー
O	CR/0	CR/1	CR/1	エラー
R	CR/1	CR/1	エラー	エラー
S	CR/1	CR/1	CR/1	-/1
I	---	---	---	---

共有バスの共有装置を制御する

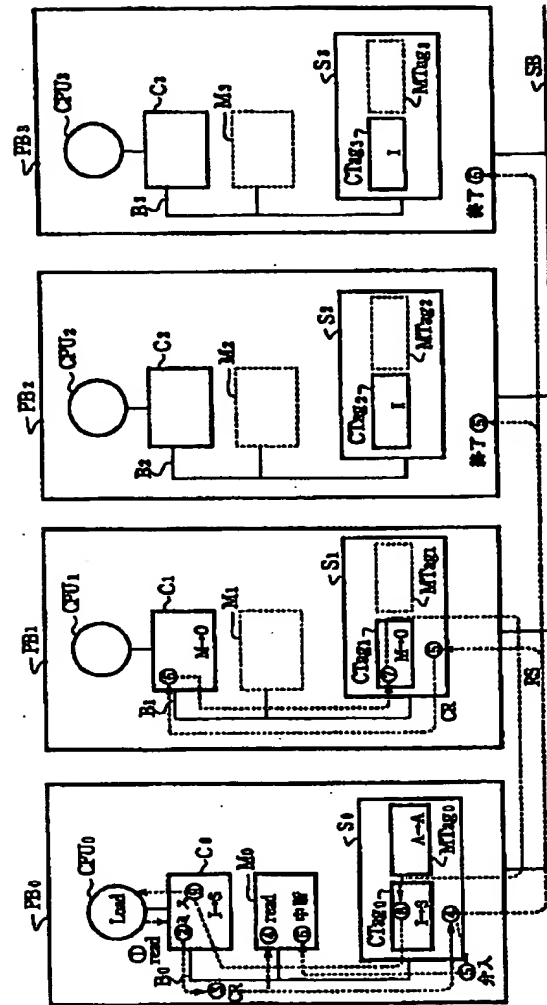
[図43]



[図44]



[図45]



[図46]

